# গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আগেও বলেছি, আরো একবার মনে করিয়ে দিই, গ্নু-লিনাক্স ইশকুলের এই পাঠমালাটা কোনো টেক্সট বই না, এটা পড়ে আপনি কিছুই শিখবেন না, শিখবেন কী করে সেটা শেখার চেষ্টা করা যেতে পারে, বড়জোর। কম্পিউটার, তাও আবার গ্নু-লিনাক্সের মত কোনো জ্যান্ত গতিশীল এবং সর্বব্যাপী বিষয় নিয়ে, আমার যোগ্যতার কয়েক কদম বাইরে। আর টেক্সট বই ব্যাপারটার প্রতি একটা অ্যালার্জি আছে আমার জন্মের থেকে, এবং নিয়েই মরতে যাব বলে বিশ্বাস। আমি নিজে গ্নু-লিনাক্সে উত্তেজিত হয়েছিলাম, যখন এলোপাথাড়ি শেখার চেষ্টা করে চলেছিলাম, তখন চিন্তার কিছু কাঠামো কিছু খোঁচ কিছু ভাঙচুর কিছু নির্মাণ নিজের কাছে খুব গুরুত্বপূর্ণ লেগেছিল, যে এলাকাগুলো আর কাউকে, যেও একইভাবে, একই রকম গোলগোল এবং অশিক্ষিত উপায়ে চেষ্টা চালাচ্ছে, সাহায্য করতে পারে নিজের চিন্তাকে নিজের কাছে গুছিয়ে নিতে, নিজের চেষ্টার সঙ্গে নিজেরই কথোপকথনে। এখানে যে লিখছে যে পড়ছে তাদের মধ্যে কমন হল ওই অপারগতা এবং গোলগোল অশিক্ষা। আর অপরে মার্জিন অফ মার্জিন নিয়ে লেখার পর থেকেই এই ইনফর্মাল লেখার প্রতি আমার একটা ভালো লাগা তৈরি হয়, আমার অনেক বন্ধুরই সেটা কাজে লেগেছিল। পলিটিকাল ইকনমি আর সংস্কৃতি স্টাডিজ হয়ত আমার নিজের জায়গা, কম্পিউটারটা নয়। কিন্তু সেই না-হওয়াটাই এই বইটার শর্ত, যেমন সত্যিকারের গরিব ঘরের কষ্টে থাকা মেয়েদের, এলিট ঘরের ফেমিনিজম বেচা সেলসগার্লদের কথা বলছি না, কিছু জায়গা থাকেই যা অন্য একটা মেয়ের জায়গা থেকে বোঝা বেশ সহজ, তেমনি আপনার আর আমার একদেশতার সূত্র এটাই — দুজনেরই না-জানা, এবং জানতে-চেষ্টা-করা। এর যেকোনো একটা শর্তকে তুলে নিলেই এই লেখাটা জাস্ট হেজে যায়। আমার খোঁজাটাকে আমি আপনার সঙ্গে শেয়ার করছি, আমি চাইছি চিন্তার ধাঁচটাকে আমি যেভাবে পেয়েছি আপনার কাছে পৌঁছে দিতে। এটা আপনার উপরেও একটা শর্ত চাপিয়ে দিচ্ছে, যদি নিজের কোনো খোঁজা এবং চেষ্টা, এই লেখা থেকে আলাদা করে, না-থাকে, আপনার এটা পড়ার কোনো মানে নেই। আপনি এটা পড়ে যে কিছুতেই গ্নু-লিনাক্স শিখবেন না, এ গ্যারান্টি আমি আগে থেকেই দিয়ে দিচ্ছি। আজ আমরা শুরু করব কয়েকটা অত্যাবশ্যক কমান্ড দিয়ে, তারপর যাব ফাইল সিস্টেমের আলোচনায়।

।। দিন ছয় ।।

১ ।। কয়েকপিস কমান্ড

এবার একদম কেজো একটা জায়গা। পাঁচ নম্বর দিনে লগ-ইন করে আপনি ঢুকে পড়েছেন সিস্টেমে, সেখানে শেল এবং কারনেলের উপস্থিতি সম্পর্কেও একটা আলগা আন্দাজ তৈরি হতে শুরু করেছে। কিন্তু এবার সিস্টেমটাকে নিজের সামনে চলমান দেখতে গেলে কয়েকটা কমান্ড জেনে নেওয়া খুব জরুরি। আমরা কয়েকটা কমান্ডের উল্লেখ করেছি এর মধ্যেই এক নম্বর দিনে দু-চারটে, এবং পাঁচ নম্বর দিনে গোটাকয়েক। সেগুলোকেই এবার একটা প্রথানিষ্ঠ রকমে ঝালিয়ে নেওয়া যাক, হ্যাঁ দেখবেন, ধূপটা যেন না-নেভে, উঁহু আওয়াজ নয়, এখানে কমান্ডপাঠ হচ্ছে ভাই।

আমাদের অভ্যস্ত গ্নু-লিনাক্স কমান্ডের কাঠামোটা অনেকটা এইরকম — মোট কমান্ড লাইনটার দুটো অংশ, একটা মূল কমান্ড, আদেশটা। অন্যটা হল তার আর্গুমেন্ট, বিধেয়। এই বিধেয়-তে আবার দুটো অংশ, একটা অপশান বা পছন্দ। অন্যটা হল যার উপর আদেশটা চলবে সেই অবজেক্ট বা কর্মটা। ধরুন আপনি পাঁচ নম্বর দিনের আপনার ওই প্রবন্ধের বইয়ের ডিরেক্টরিতে ঢুকেছেন। একটু আত্মশ্লাঘা পেতে চাইছেন নিজের প্রবন্ধরাজির গাবদা গাবদা গতর দেখে। এই মুহূর্তে আপনি যে প্রবন্ধটা লিখছেন তার নাম essay.4.text — আগেরগুলো শেষ হয়ে গেছে। অবভিয়াসলি, আমারও তাই। পরশু প্লাস্টার কাটবে, ঠ্যাং ছুটি শেষ হয়ে এল, আজ নয়ই ডিসেম্বর, এখন প্রায় যুদ্ধকালীন তৎপরতায় লিখতে হচ্ছে।

এবার, আপনি এর সাইজ জানবেন, কমান্ড দিলেন —

```
ls -s -h essay.4.txt
```

এবং এর পর এন্টার মারলেন, অহো, কী আনন্দ, বাইট সাইজ দেখছি ধ্যাড়াধ্ধ্যাড় করে বেড়ে যাচ্ছে, অহো, কী আনন্দ, মনে হচ্ছে এই ধরিত্রীতে একটা পদচিহ্ন না রেখে কিছুতেই মরব না, বিগলিত হয়ে গেলেন ফাইলের সাইজ দেখে। এবার কমান্ডের কাঠামোটা দেখুন। এর এই '`ls`' অংশটা হল মূল কমান্ড বা আদেশ, '`-s -h`' অংশটা হল অপশান বা পছন্দ, '`-s -h essay.4.txt`' এই পুরোটা হল আর্গুমেন্ট বা বিধেয়, এবং '`ls -s -h essay.4.txt`' পুরোটা মিলিয়ে হল গোটা কমান্ডলাইনটা। এখানে আপনার হয়ে পছন্দটা আমি করে দিয়েছি, বলে দিয়েছি কী অপশান হবে। '`s`' মানে সাইজটা দেখাও, আর '`h`' অপশানটা বলে দিচ্ছে ওই সাইজের একক করো বাইট বা কিলোবাইট বা মেগাবাইট এমন কিছু যা মানুষে বোঝে, হিউম্যান রিডেবল হয়, নয়তো ও সাইজটা দেবে ব্লক-এর সংখ্যা দিয়ে। ব্লক ব্যাপারটায় আসছি আমরা একটু বাদে। দুটো অপশানেরই আগে, দেখুন, আমরা

একটা করে '-' চিহ্ন দিয়েছি। এটা অনেক কমান্ডের সঙ্গে দিতে হয়, অনেক কমান্ডের সঙ্গে দিতে হয়না। অনেক কমান্ডের বেলায় আবার দুটো অপশানকে একসঙ্গে জুড়ে দেওয়া যায়। যেমন এই কমান্ডটা হুবহু একই ভাবে কাজ করত যদি আমরা কমান্ডলাইন দিতাম — `ls -sh essay.4.txt`। এবার প্রশ্ন হল, কোনটার উপর কমান্ডটা কাজ করবে, মানে কর্ম বা অবজেক্ট, এই উদাহরণে যা '`essay.4.txt`', সেটা নয় আপনি জানেন, এবং কী কমান্ড সেটাও জেনে ফেললেন, কিন্তু কোন কমান্ডে কী অপশান, কোন অপশান কী আকারে দেওয়া যায়, এই হাজারো খুঁটিনাটি আপনি কেমন করে জানবেন? জানবেন এই টেক্সট থেকে।

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
-a, --all                do not hide entries starting with .
-h, --human-readable     print sizes in human readable format (e.g., 1K 3M 2G)
-i, --inode              print index number of each file
-l                       use a long listing format
-R, --recursive          list subdirectories recursively
-s, --size               print size of each file, in blocks
-t                       sort by modification time
-1                       list one file per line
    --help               display this help and exit
```

এটা অনেকটা ত্র্যম্বকের ত্রিনয়ন ত্রিকাল ত্রিগুণের মত হল, এই টেক্সটা গজাল কোত্থেকে? ধীরে। এই টেক্সটের গজানোর ইতিহাসটা এই টেক্সটের মধ্যেই আছে। দেখুন তো আলোচনাটা হওয়ার আগে সেটা নিজেই খুঁজে পান কিনা? পেলেন? প্রথম লাইনটা দেখুন, ব্যবহারবিধি বা ইউসেজ, ঠিক অপশান আর্গুমেন্ট কমান্ড-লাইন নিয়ে আমরা এর আগেই যা বললাম সেই কথাটাই অন্য আর এক রকমে বলা। দ্বিতীয় লাইনে বলা কী কাজ করে এই কমান্ডটা। তিন থেকে এগারো নম্বর অব্দি প্রতিটি লাইনের কাঠামোই এক। প্রথমে একটা অপশানের ছোট আকার, তারপর বড় আকার, তারপর সেই অপশানটা কী কাজ করে। ছোট আর বড় দুটোই দেওয়া যায়, ছোটটা দেওয়ার সুবিধে, আর বড়টা মনে রাখার। শুধু দেখুন ছয়, নয় আর দশ নম্বর লাইনের পছন্দের কোনো বড় আকার নেই, শুধু ছোটটাই আছে। আর এগারো নম্বর লাইনের পছন্দটার শুধু বড় আকারটাই আছে। এর মধ্যে তিন নম্বর লাইনের প্রথম অপশানটা বুঝতে গ্নু-লিনাক্সে কম পরিচিতদের একটু অসুবিধে হতে পারে। একটা ডিরেক্টরির সব ফাইলকে স্বাভাবিক অবস্থায় `ls` দেখায় না। যেসব ফাইল বা ডিরেক্টরির নামের আগে একটা করে '.', বিন্দু বা ডট আছে, যেমন ধরুন .kde বা .mplayer ইত্যাদি। সচরাচর এই ফাইলগুলোয় থাকে সিস্টেম সংক্রান্ত কিছু যা দৈনন্দিন কাজে দরকার পড়েনা একজন ব্যবহারকারীর। এই প্রথম অপশানটা, '`-a`', দিলে `ls` সেই ফাইলগুলোকেও দেখায়। চার আর আট নম্বর লাইনের অপশানদুটো তো আমরা এইমাত্র ব্যবহার করলাম। পাঁচ নম্বর লাইনের অপশানটা একটা ফাইলের ইনডেক্স নোড বা আইনোড দেখায়, সেটার কথায় আমরা আসছি একটু বাদে। ছ নম্বর লাইনের '`-l`' অপশানটা লম্বা লাইনে বড় করে ফাইল সংক্রান্ত অনেকটা তথ্যকে হাজির করে, খুবই কাজে লাগে ফাইলের মালিকানা এবং ব্যবহারযোগ্যতা বুঝতে। পরে দেখব আমরা। '`-R`' বলে, যে ডিরেক্টরিতে বসে `ls` মেরেছি আমরা সেই ডিরেক্টরির মধ্যে যদি কোনো সাবডিরেক্টরি থাকে, তার মধ্যে আরো আরো, সেই ডিরেক্টরিগুলোকেও একই ভাবে দেখিয়ে যেতে। '`-t`' ফাইলগুলোকে সাজায় অক্ষরের অনুক্রমে নয়, তাদের গায়ের সময়চিহ্নের নিরিখে। '`-1`' বলে প্রতিটি ফাইলের জন্যে একটা করে লাইন বরাদ্দ করতে।

আর '`--help`' হল সেই অপশান যা দিয়ে '`ls --help`' কমান্ডটা মেরে আমি এই টেক্সটাকে স্ক্রিনে পেয়েছিলাম। তারপর '`ls --help > ls.help.text`' মেরে একটা ফাইল বানিয়েছিলাম যার নাম ls.help.text । এই ভাবে ফাইলে পাঠানোটাকে কী বলে — মনে আছে আপনার? সেই ফাইল থেকে এখানে তুলে দিলাম। যদিও, অসভ্য ভাবে ছাঁটকাঁট করে, ঠিক যতটুকু আমার মনে হয়েছে আমাদের আলোচনার জন্যে এই মুহূর্তে প্রয়োজনীয় শুধু সেটুকুই রেখেছি। কতটা ছাঁটা বলুন তো? ৯৭ লাইনকে ১১ লাইন বানানো। গোটাটা পড়ার ইচ্ছে হলে, কোনো ব্যাপার না, গ্নু-লিনাক্স সিস্টেম আপনার জন্যে অবারিতদ্বার, বলা মাত্র বানিয়ে দেবে।

আর বলতেও হবেনা, বানানোই আছে। এই হেল্প ফাইলটা যে ফাইলের একটা কুচো। ঠিক পাসওয়ার্ড নিয়ে যেটা করেছিলাম পাঁচ নম্বর দিনে — মনে আছে? '`man 5 passwd`' বলে একটা কমান্ড ব্যবহার করেছিলাম। এই 'man' হল ম্যানুয়াল। '`ls`' কমান্ডটা বিষয়েও সবচেয়ে ভালোভাবে জানার উপায় হল '`man ls`'। শুধু '`ls`' কেন, যেকোনো কমান্ডের বেলাতেই তাই। কিন্তু এই '`man`' ব্যবহার করতে গেলে এই কমান্ডটাকে তো বুঝতে হবে। কেন, তার জন্যে আছে ম্যানুয়ালের ম্যানুয়াল। '`man man`' কমান্ড দিয়ে আমরা 'man' সম্পর্কে জানতে পারব। আর ওই '`--help`' পছন্দটা তো আছেই। ওটা জুড়ে দিয়ে যে কোনো কমান্ড দেওয়া মানেই তার একটা সংক্ষিপ্ত হেল্প ফাইল পেয়ে যাওয়া, যারই বড় ভাই হল ম্যান বা ম্যানুয়াল। এই গোটা ম্যানুয়ালের পাতার সমগ্রতাটা ভরা থাকে গ্নু-লিনাক্স সিস্টেমের ভিতরেই। আপনার অঙ্গুলি-পীড়নের অপেক্ষায়। কিবোর্ডে

টাইপ করে কমান্ড দিলেই হল। এই ম্যান বা ম্যানুয়াল পাতাগুলোর আবার কয়েকটা সেকশন আছে। সেই সেকশন অনুযায়ী আবার খোঁজা যেতে পারে বিশেষ বিশেষ কমান্ডকে। এর সবটুকু এই মুহূর্তে আপনার কাছে স্পষ্ট হবেনা, হতে থাকবে যত আপনি আপনার সিস্টেমটা ব্যবহার করবেন।
এই সেকশনগুলো হল ১, এক্সিকিউটেবল বা চালানো যায় এমন প্রোগ্রাম বা শেল কমান্ড। ২, সিস্টেম কল বা কারনেলের ফাংশন। ৩, লাইব্রেরি কল বা সিস্টেম লাইব্রেরিগুলোর কোনো ফাংশন। ৪, বিশেষ ধরনের কোনো ফাইল — /dev ডিরেক্টরিতে যেমন পাওয়া যায়। ৫, ফাইলের আকার আকৃতি এবং লেখার নিয়ম — /etc/passwd ফাইলে যেমন। ৬, গেমস। ৭, বিভিন্ন ম্যাক্রো প্যাকেজ এবং তার নিয়ম — যেমন man(7) বা groff(7)। ৮, সিস্টেম অ্যাডমিন্সট্রেশন কমান্ড — যাদের শুধু রুট ব্যবহার করতে পারে। আর ৯, বিশেষ ধরনের কারনেল রুটিন। এই সেকশনগুলো পড়ে যদি আপনার এরকম মনে হয়, এগুলো খায় না মাথায় দেয়, নো ফিয়ার, আমি যখন লিখতে পারছি, খায় না মাথায় দেয় এর চেয়ে খুব বেশি কিছু না-বুঝেই, আপনি তখন আলবাত এটা পড়তে পারেন ওটুকু বাদ দিয়েই। তৃতীয় বিশ্বের নাগরিক, আমাদের মস্তিষ্কে ডবল খাঁজ, সাহেবের বাচ্চাগুলো যখন একটা ভাষায় সব কিছু করতে পারে, আমাদের তখন গোটা তিনেক ভাষা তো জানতেই হয়, কী জানব আর কী জানব না এটা জানতে গিয়েই। তাই একসঙ্গে সবটা বুঝে যাবেন নিখুঁত ভাবে এটাই একটা সাহেবী কুসংস্কার। যখন দেখবেন, জানেন কি জানেন না এটা একটু ঘোলাটে, কিন্তু মোটামুটি একটা আন্দাজ পেয়ে গেছেন, কী কাজ বা কী ভাবে করতে হবে, তখনই বুঝতে হবে বুঝে গেছেন। শুধু একটা কথা বলুন তো, এর ৮ নম্বরটা, সিস্টেম অ্যাডমিন্সট্রেশন কমান্ড, যাদের শুধু রুট ব্যবহার করতে পারে, সেই কমান্ডগুলোও তো এক একটা প্রোগ্রাম, এক একটা এক্সিকিউটেবল বাইনারি ফাইল, যাদের নাম লিখে এন্টার মারলে চলে, সেই ফাইলগুলো কোন বা কোন কোন ডাইরেক্টরিতে আছে? পাঁচ নম্বর দিনের ১১ নম্বর সেকশনে, রুটের এর বা পথনির্দেশটা দেখে দেখুন তো বলতে পারেন কিনা? না পারলে, মাইরি বলছি, আপনার জন্যে লিখতে আমার বেশ বোর লাগছে। যাকগে, বেনিফিট অফ ডাউট পেয়ে গেলেন, এতো আর জিএলটি ইশকুলের ক্লাস নয়, যার নোট হিশেবে এই বইটা লেখার শুরু, যে সামনেই আছেন। ধরে নিলাম বলতে পেরেছেন। ২ নম্বর সেকশন মানে সিস্টেম কল নিয়ে কিছু আলোচনা আছে এক আর দুই নম্বর দিনে। আর তিন মানে লাইব্রেরি নিয়ে আলোচনা ছিল তিন নম্বর দিনে, আডার কম্পিউটার ভাষা নিয়ে ভাবনাচিন্তার প্রসঙ্গে। দেখে নিতে পারেন।
সিস্টেমের মধ্যে আপনার সাহায্যার্থে শুধু যে এই ম্যান ভরা আছে তাই নয়, আছে আরো একটা জিনিষ, তার নাম ইনফো —

```
info --help
Usage: info [OPTION]... [MENU-ITEM...]
Read documentation in Info format.
Options:
      --apropos=STRING         look up STRING in all indices of all manuals.
  -d, --directory=DIR          add DIR to INFOPATH.
      --dribble=FILENAME       remember user keystrokes in FILENAME.
  -f, --file=FILENAME          specify Info file to visit.
  -h, --help                   display this help and exit.
      --index-search=STRING    go to node pointed by index entry STRING.
  -n, --node=NODENAME          specify nodes in first visited Info file.
  -o, --output=FILENAME        output selected nodes to FILENAME.

Examples:
  info                       show top-level dir menu
  info emacs                 start at emacs node from top-level dir
  info emacs buffers         start at buffers node within emacs manual
  info --show-options emacs  start at node with emacs' command line options
  info -f ./foo.info         show file ./foo.info, not searching dir
```

`info`। এখানে দেখুন, আমরা সেই ইনফোর হেল্প ফাইল, মানে '`info --help > info.help.text`' মেরে তৈরি, তার থেকে কয়েকলাইন তুলে দিচ্ছি। এবার আর ব্যাখ্যা করছি না, দেখুন তো, আপনি নিজেই পারেন কিনা। ম্যান কমান্ডের খুঁটিনাটি এবার নিজেই পড়ে নিন, '`man man`' মেরে। দু-একটা খুব কাজের কথা বলে দিই। ম্যান দিয়ে যেকোনো কমান্ডের ম্যানুয়াল পেজ খুলে পড়বেন যখন স্পেসবার টিপলে এগোবেন, 'বি', B, সুইচটা টিপলে পিছোবেন, আর 'কিউ', Q, সুইচটা টিপলে বেরিয়ে যাবেন। এটা অভিমন্যুকে দেখে শিখুন, কোথাও ঢোকার আগে মনে রাখবেন বেরোনোর উপায়। কোনো কিছু খুঁজতে হলে লাগবে 'স্ল্যাশ' বা / সুইচটা। এটা টেপা মাত্র দেখবেন একদম নিচে এটা ফুটে উঠেছে, সেখানে টাইপ করে দিন আপনি যা খুঁজতে চান, এবং এন্টার মারুন। এই চারটে সুইচ শুধু এই ম্যান না, আরো বহু গ্নু-লিনাক্স কমান্ডের বেলাতেই সত্যি। এবার ধরুন, আপনি যা খুঁজতে চাইছেন সেটা কোনো কমান্ড নয়, একটা কোনো বিশেষ শব্দ, সেটা কোথায় কোথায় আছে, কী

আছে, এটা আপনি জানতে চাইছেন। ধরুন কথাটা হল CD, সিডি নিয়ে আপনার সিস্টেমের ম্যানুয়াল পেজগুলোয় কী বলা আছে আপনি জানতে চাইলেন। আপনি কমান্ড দিন 'man -k CD', দেখুন স্ক্রিনে একটা তালিকা ফুটে উঠেছে। গোটা তালিকাটার প্রত্যেক লাইনেই বাঁ দিকে একটা শব্দ, তারপর ব্রাকেটে একটা নম্বর, তারপর ডানদিকে লেখা বাঁদিকের ওই শব্দটা কী তাই নিয়ে একটা বিবৃতি। যেমন আমার সুজে ৮.২ সিস্টেমে আমি 'man -k CD' করে যে তালিকাটা পেলাম সেটার একটা লাইন —

```
cdrecord (1)         - record audio or data Compact Discs from a master
```

একদম বাঁদিকে সিডিরেকর্ড বলে একটা কমান্ডের মানে একটা এক্সিকিউটেবল প্রোগ্রামের নাম, তারপর ব্রাকেটে লেখা (১), কারণ, একটু আগের ম্যানুয়াল পেজগুলোর সেকশনের তালিকাটা দেখুন, ১ নম্বর সেকশনে থাকার কথা এক্সিকিউটেবল প্রোগ্রাম বা শেল কমান্ডগুলোর। এবং ডানদিকে লেখা সিডিরেকর্ড সম্পর্কে একটা কুচো মন্তব্য। সিডিরেকর্ড বস্তুটা ঠিক কী করে। এবার আপনি যদি এই সিডিরেকর্ড সম্পর্কে জানতে চান তাহলে কমান্ড দেবেন 'man 1 cdrecord'। এবার বলতে পারেন ওই কমান্ডটায়, man 5 passwd, আমরা ১ না দিয়ে ৫ দিয়েছিলাম কেন? সেকশনের তালিকার সঙ্গে মিলিয়ে দেখুন।

এখানে একটা ঘাপলা আছে। ঘাপলাটাকে একদম শাঁসেজলে ফিল করতে চাইলে এই কমান্ডটা টাইপ করুন, 'man -k X', এবার আপনার প্রসেসর যদি ওইসব চৌষট্টি বিট ফিটের মালেদের একজন না হয়, আপনি স্বচ্ছন্দে একটা সিগারেট ধরান। কারণ, বেশ কিছু সময় ধরে এক স্ক্রিন আপনি এক পলকের বেশি দেখতে পাবেন না। পাতার পর পাতা চলতেই থাকবে। এই 'X' মানে মূলত এক্স-উইনডোজ বা গ্রাফিকাল ইউজার ইন্টারফেস বা গুই নিয়ে আলোচনা আছে আমার সুজে সিস্টেমে মোট ২৮২৯ খানা ম্যানুয়াল নিবন্ধে। সিস্টেম ম্যানুয়ালের মোট সাইজ বুঝতে পারছেন? একটু আগে 'CD' খুঁজেছিলাম যখন তখনো তার সংখ্যা পেয়েছিলাম ৩৮ খানা, তার মানে এক স্ক্রিনের মানে ২৫ লাইনের বেশি। তাহলে খুঁজবেন কী করে? শুধু ডিসপ্লেটা যখন থেমে যাবে — মানে তালিকার শেষ পাতায়? এটা একটা স্ট্যান্ডার্ড কথা হল? লেস আছে কী করতে? লেস নিয়ে একটা মজার কথা চলে, লেস ইজ মোর দ্যান মোর। মানে, মোর বলে একটা পেজার আগেই ছিল, সেই পুরোনো মোর-এর সঙ্গে আরো কিছু কাজ এনে তৈরি করা হয় এই লেস বলে পেজারটাকে। পেজার বলতে একটা সফটওয়ার যা কোনো টেক্সটকে পাতা বাই পাতা পরপর দেখিয়ে যায়। এবার আর স্ক্রিন একটি নদীর নাম, স্বপ্নের ছন্দে সে বহিয়া যায় — তা নয়, তাকে বেঁধে ফেলা হয়েছে লেস-এর শেকলে। লেস-এর পোষাকের অনির্বচনীয়তা নিয়ে আপনার চোখে হয়তো ধরা দেবে না, কিন্তু কাজ ভারি ভালো চলে যাবে।

man -k CD | less — এটায় আমরা মূল কমান্ডের আউটপুট বা ফলাফলটাকে পাইপ করে দিলাম লেস দিয়ে। এবার ধীরে সুস্থে পাতাটাকে নামান। ঠিক একটু আগে ম্যান পেজের বেলায় যে চারটে সুইচ বলেছিলাম, স্পেসবার, বি, কিউ এবং স্ল্যাশ — সেই চারটেই একই কাজ করে এখানেও। শুধু এখানে না, ইনফো-তেও করবে, আরো অনেক কমান্ডেও। এবার বলুন তো, মোট কত লাইন ছিল ম্যান-এর ফলাফলে, ২৮২৯, যে বললাম, বললাম কী করে? গুণে গুণে? প্লিজ, আমার বাবার দিককার কাস্ট কায়স্থ, মানে ক্লার্ক, কিন্তু অতো মাছি মারার প্র্যাকটিস আমার কোনোদিনই নেই।

এবার দেখুন, লেস করেও যদি খুঁজতে চান, বহু বহু পাতা অনেক ক্ষেত্রেই, তখন খুঁজে পাওয়াটা অসম্ভব আপনি যা খুঁজছেন। তখন অন্য একটা তরকিব কাজে লাগান। তার নাম গ্রেপ। ধরুন, আপনি সে সিডি নিয়ে খুঁজছেন সেটা আসলে আপনার অডিয়ো সিডি নিয়ে কোনো কৌতূহলে। তাহলে এবার ম্যান-এর আউটপুটটাকে পাইপ করে দিন গ্রেপ-এ।

man -k CD | grep 'audio' — এর থেকে স্ক্রিনে যা পাবেন সেটা এই পাতার চওড়ায় পাশাপাশি ধরানোর জন্যে একটু কেটেছেঁটে অনেকটা এইরকম —

```
cdda2wav (1)     - a sampling utility that dumps CD audio data into wav files
cdrecord (1)     - record audio or data Compact Discs from a master
cdparanoia (1)   - an audio CD reading utility with extra data verification
cdrdao (1)       - writes audio CD-Rs in disc-at-once mode
cda (1)          - Compact disc digital audio player utility
xmcd (1)         - CD digital audio player utility for X11/Motif
```

এবার ধরুন, এইটা যখন স্ক্রিনে দেখছেন, হঠাৎ খেয়াল করলেন, আরে, চারপাশটা এত নিস্তব্ধ লাগছে কেন, স্ক্রিনে 'date' মেরে বা ঘড়িতে দেখলেন, আরে আড়াইটে বেজে গেছে, ধুর ছাতা, এত রাতে অডিয়ো সিডির এত প্রোগ্রাম খুঁজে পেয়েই বা কী হল, বাজানো তো যাবেনা, পাড়ার আগেও বাড়ি, তারপর কালকের অফিস, একটা দীর্ঘশ্বাস ফেলে আপনি এই লাইনকটা তুলে রাখতে চাইলেন একটা ফাইলে, কালকে যাতে এত খোঁজাখুঁজি করতে না হয়, আপনি কমান্ড দিলেন

man -k CD | grep 'audio'> audio.cd.program.text, তারপর একবার শেষ দেখে নিতে চাইলেন, ফাইলটা ঠিক সেভ হয়েছে কিনা, বা তার সাইজ ঠিক কতটা। 'ls -sh audio.' টাইপ করে দিলেন স্ক্রিনে, কিন্তু সাবধান, এবার কিন্তু এখনো আপনি এন্টার সুইচটা মারেননি। তার জায়গায় মারবেন ট্যাব সুইচটা। এটা ব্যাশ শেলের কমান্ড কমপ্লিশন

বা নিজে থেকে আদেশের শূন্যস্থান পুরন করে দেওয়ার কাজ। যদি আপনার ওই ডাইরেক্টরিতে আর কোনো ফাইল না থাকে যার নাম 'audio.' দিয়ে শুরু, তাহলে ইতিমধ্যেই আপনার স্ক্রিনে জ্বলজ্বল করছে আপনার গোটা দিতে চাওয়া কমান্ডটা, 'ls -sh audio.cd.program.text'। আর যদি একাধিক ফাইল থাকে যারা প্রত্যেকেই 'audio.' দিয়ে শুরু, তাহলে স্ক্রিনে পাবেন আপনার কমান্ডটা যতটা আপনি টাইপ করেছেন, তার নিচে ওই কটা ফাইলের নাম। যে অব্দি গেলে এই নামটা সম্পূর্ণ নিজস্ব চেহারা পেয়ে যাবে সেই অব্দি আপনি টাইপ করে দেবেন, তারপর গোটাটুকু ভরে দেবে ব্যাশ নিজেই। যেদিন আপনার একটা দুটো বর্ণ টাইপ করার পরই আপনার বাঁ হাতের কড়ে আঙুল ট্যাব সুইচের দিকে যেতে শুরু করে দেবে, সেদিন বুঝবেন আপনি ব্যাশ শেলে অভ্যস্ত হতে শুরু করেছেন। ধরুন, আপনি 'ls -sh audio.' টাইপ করার পর দেখলেন স্ক্রিনে ফুটে উঠেছে আপনার টাইপ করা কমান্ডটুকু এবং নিচে দুটো ফাইলের নাম, 'audio.cd.program.text' এবং 'audio.filetypes.list', এবার আপনি জাস্ট একবার 'সি' সুইচটা টিপে ট্যাব মারলেই ব্যাশ গোটা কমান্ডটা ফুটিয়ে তুলবে, কারণ, 'সি'-র থেকে দুটো ফাইল আলাদা, ওই 'সি'-টা আছে শুধু আপনার দেখতে চাওয়া ফাইলটাতেই। এবার যেই এন্টার মারবেন, স্ক্রিনে ফুটে উঠবে ফাইলটার নাম এবং মানববোধ্য সাইজ, আগেই তো বলেছি। সাইজটা দেখার পর নিজের কৃতিত্বে মুগ্ধ আপনার মনে হল, আরে পরের বার এতোটা টাইপ যাতে না করতে হয়, ব্যাশের কমান্ড কমপ্লিশন ফাংশনটার পুরো মজাটা যাতে পাওয়া যায়, তার জন্যে 'audio.filetypes.list' ফাইলটার নামটা তো বদলে দিলেই হয়। আপনি অন্য একটা নামে, 'sound.filetypes.list', ফাইলটা কপি করবেন। কপি করার কমান্ড 'cp' তো আমরা আগে থেকেই জানি, পাঁচ নম্বর দিনে তিন নম্বর সেকশনে ছিল।

কমান্ড দিলেন, 'cp audio.filetypes.list sound.filetypes.list'। এবার এন্টার মারলেন, কমান্ড প্রম্পট সঙ্গে সঙ্গে চলে এল, মানে কাজটা হয়ে গেছে। কাজটা না হলে জানাতে পারে, চুপচাপ ফিরে এল মানে কাজ সমাধা, শেল এভাবেই কাজ করে। এবার একবার 'ls' মেরে দেখে নিলেন। দেখতে গিয়েই মনে হল, ও হরি, পুরোনো 'audio.filetypes.list' ফাইলটাও তো রয়ে গেল। কোই পরোয়া নেই ফাইল ওড়ানোর কমান্ড 'rm' — সেটাও তো আপনি জানেন। সহজেই উড়িয়ে দিলেন ফাইলটা। 'rm audio.filetypes.list'। এন্টার মারলেন। প্রম্পট ফেরত এল। মানে ফাইল উড়ে গেছে। একবার 'ls' মেরে দেখে নিলেন। তারপর মাথায় এল, আরে প্রথমেই কপি করে, তারপর আরএম বা রিমুভ না করে, একবারেই তো করা যেত 'mv' দিয়ে, সেটা তো ফাইলকে নড়ায় বা নাম বদলায়। তখন একটা কমান্ডেই হয়ে যেত — 'mv audio.filetypes.list sound.filetypes.list'। যাকগে।

শুতে যাওয়ার আগে এখনো একটা কাজ বাকি আছে আপনার। আপনি একবার দেখে নিতে চাইলেন, ফাইলটা ঠিক কোথায় আছে, আপনার হোম ডিরেক্টরির কোথায়? কোন সাবডিরেক্টরিতে? এর জন্যে আপনি যে কমান্ডটা ব্যবহার করবেন সেটা হল 'pwd'। মানে, কোন ডিরেক্টরিতে কাজ করছি সেটা দেখাও। প্রিন্ট ওয়ার্কিং ডিরেক্টরি। এর পরে আপনি যে ডিরেক্টরিতে সেভ করেছেন ফাইলটা, তার ঠিকানাটা ফুটে উঠবে। ধরুন এটা আমি করছি। আমি ফাইলটা সেভ করেছি গার্বেজ বলে একটা ডিরেক্টরিতে। এই গার্বেজ ডিরেক্টরিটা, ধরুন অডিয়ো নামের ডিরেক্টরিতে। যে অডিয়ো ডিরেক্টরিটা আছে আমার হোম ডিরেক্টরির মাল্টিমিডিয়া বলে একটা ডিরেক্টরিতে। এই অবস্থায় আমি 'pwd' মারলে স্ক্রিনে ফুটে উঠবে —

/home/dd/multimedia/audio — এটাই এই মুহূর্তে আমার কাজের ডিরেক্টরি। কাল এসে আর খুঁজে পেতে এবং অডিয়ো সিডির গান বাজাতে অসুবিধে হবেনা।

পাঁচ নম্বর দিন আর আজকের শুরু থেকে আমরা যে ষোলটা কমান্ড এখনো পেয়েছি সেগুলো হল — mkdir, rmdir, cd, rm, mv, cp, ls, who, tar, less, wc, man, bzip2, info, grep, pwd। এর মধ্যে আবার mkdir, rmdir, cd কমান্ডগুলো আলোচনায় আনতে পারিনি কারণ এখনো আমরা রিলেটিভ এবং অ্যাবসলিউট পাথ, আপেক্ষিক এবং চূড়ান্ত পথনির্দেশ বলতে কী বোঝায় সেই প্রসঙ্গ এখনো আনতে পারিনি। mv কমান্ডটাকে নড়ানোর কাজে কী ভাবে ব্যবহার করা যায় সেটাও বলা হয়নি সেই একই কারণে। সেই পাথ বা পথ-এর কথা বলতে গেলে একটু ফাইল সিস্টেম জানতেই হবে। এবার আসবে। প্রথমে ফাইল, তারপর ডাইরেক্টরি সিস্টেম, তারপর ফাইল সিস্টেম। কিন্তু যা বলেছিলাম, এই লেখাটায় আসলে আমরা দুজনেই খেলছি, নইলে খেলাটাই হবেনা। আপনি এখন যতদূর সম্ভব তার চেয়ে একটু বেশি পরিশ্রম করে এই কমান্ডগুলোয়, এদের ম্যান আর ইনফোয় অভ্যস্ত হয়ে নেবেন আপনার সিস্টেমে। যত পারা যায় নানা আলাদা আলাদা অপশান কমান্ডের সঙ্গে লাগিয়ে লাগিয়ে দেখবেন তাতে কী হয় কী না হয়। আর যতটা মারেন ম্যানুয়াল পড়া অভ্যেশ করে ফেলুন এখন থেকে। সামনে দুঃখের দিন আসছে। এখানে যত বেশি পরিশ্রম করবেন, তত আপনার পরে সুবিধে হবে। অনেকটা গানের রিওয়াজের মত। টাচ-টাইপিং বা দশ আঙুলে টাইপ করার মত। এই টাচ টাইপিং-এর জন্যে একটা চমৎকার গ্নু সফটওয়ার আছে, জিটাইপিস্ট। পরে আমরা আসব সেসবে। এখানে হঠাৎ একটা কথা মাথায় এলো, বারবার আ মি লিখে যাচ্ছি পেজ আপ পেজ ডাউন করে পড়ার কথা, তার মানে আমি কি ধরে নিচ্ছি যে আপনি এটা কম্পিউটারে পড়ছেন? যদি তাই হয় তাহলে সেটা কী? উইনডোজ একটা প্রতিবেশ? যদি তাই হয়, তাহলে তো আপনি এটা পড়াকালীন করে দেখতে পারবেন না। যদি এটা গ্নু-লিনাক্স প্রতিবেশ হয়,

তাহলে সেটা একই সঙ্গে একটা এক্স-উইনডোজ-ও। গুহনোম হোক, কেডিই হোক, ব্ল্যাকবক্স বা ফ্লাক্সবক্স বা অন্যকিছু যাই হোক। কারণ এটা একটা পিডিএফ ফাইল, যা প্রিন্ট করা যাবে কমান্ড লাইনে, কিন্তু কমান্ড মোডে পড়ার কোনো প্রকৌশল আছে বলে জানিনা। কার্সেস দিয়ে ফ্রেমবাফার দিয়ে কিছু কখনো তো শুনিওনি। এফবিআই দিয়ে যেমন গ্রাফিক্স দেখা যায়, বা এমপ্লেয়ার যেমন কমান্ড মোডেই দিব্য সিনেমা দেখায়। তার মানে আপনার মেশিনে গ্নু-লিনাক্সে এক্স-উইনডোজ চলছে। এই অবস্থায় আপনি একটা টার্মিনাল খুলে নিন। এবং সেখানে কমান্ডগুলো পড়তে এবং করতে থাকুন। আর সঙ্কর্ষণরা যেমন প্রাণপণ দৌড়চ্ছে এটাকে হার্ড প্রিন্টে মানে বই আকারে বার না-করে ছাড়বে না, সেটা যদি শেষ অব্দি ঘটে যায়, যদি, কারণ, একটা বিরাট পরিমাণ টাকা লাগবে বইটা বার করতে, সেটা কলকাতা লাগ-এর বা এফএসএফ-ওয়েস্টবেঙ্গলের নেই — বই আকারে যদি পড়েন তাহলে অবশ্য সমস্যা নেই, কমান্ড মোডেই করে দেখতে পারবেন বই পড়ার পাশাপাশি।

এখানে একটা প্রাক্টিকাল সমস্যার কথা বলি, যে জিএলটি ক্লাসের সূত্রে প্রথমে ক্লাসনোটস এবং তারপর ক্রমে গতরবৃদ্ধি হতে হতে ক্রমে এই বইয়ের আকারে চলে এল, তার ক্লাস চলাকালীন অশেষ সমস্যাটার কথা বলেছিল। ও তখনো কেবল সিস্টেমে ঢুকতে শুরু করেছে, তখনো চেনাজানা তৈরি হয়নি, ওর নাকি এটা খুব ধাঁধার মত লাগছ, কাদের ম্যানুয়াল পেজ পড়া দিয়ে শুরু করব? মানে কমান্ডগুলোর বিবরণ নয় ম্যানুয়াল পেজে লেখা আছে, যেমন ls কমান্ডটাকে জানলাম '`man ls`' কমান্ড দিয়ে কিন্তু এই কমান্ডটার কথাই বা জানব কী করে? মানে কাকে ম্যান দিয়ে খুঁজব? আপাতত এটুকুই বোধহয় জেনে রাখা ভালো, /bin ডিরেক্টরিতে যে যে ফাইল আছে, তাদের ম্যানুয়াল পেজ পড়া দিয়েই শুরু করা ভালো। এই /bin ডিরেক্টরিতে যারা বিরাজ করেন তারা সবাই একজিকিউটেবল মানে প্রোগ্রাম বা বাইনারি ফাইল। এর হাল হকিকত আমরা বেশ জেনে যাব আজকের আলোচনা শেষ হওয়ার আগেই। এই ডিরেক্টরির প্রোগ্রামগুলো একের পর এক ধরে ধরে ম্যান পেজ খোলা, সব সময়েই কি লোকে পড়ে নাকি, ছি, গানবাজনাসিনেমা আছে কী জন্যে, একটু নাড়াচাড়া করা, সিস্টেমের সঙ্গে ঘনিষ্ঠতা বাড়ানোর একটা ভালো রাস্তা। কারা কারা আছেন এই ডিরেক্টরিতে জানব কী করে? কেন, সিস্টেমের যে কোনো ডিরেক্টরিতে দাঁড়িয়েই '`ls /bin`' মারুন। দরকার হলে রিডাইরেক্ট করে নিন একটা টেক্সট ফাইলে, '`ls /bin > binary.file.list`' বা যে কোনো নামের একটা ফাইলে। একটা কথা মাথায় রাখুন, এই সমস্যাটাও হতে দেখেছি জিএলটি ইশকুলে, যাকে তাকে কিন্তু ক্যাট করা বা রিডাইরেক্ট করে ফাইল বানানো যায়না, মানে যায় তো সবই, একটা বাইনারি ফাইলকে যদি আপনি ক্যাট করেন বা রিডাইরেক্ট করে একটা ফাইল বানান, সেটাকে লেস দিয়ে পড়ার চেষ্টা করেন, শূন্য নম্বর দিনের ভিশুয়াল চ্যাংড়ামির গোত্রের জিনিষপত্তর পাবেন। কিন্তু ওই বাইনারি ফাইলের ম্যানুয়ালটাকে রিডাইরেক্ট করে ফাইল বানালে বেশ পড়তে পারবেন।

২.১।। ফাইল

ফাইল প্রাথমিকভাবেই একটা বিমূর্ত জিনিষ। এই গ্নু-লিনাক্স ইশকুল পাঠমালার এক নম্বর দিনে আমরা একটা সাঙ্কেতিক বা সিম্বলিক ব্যবস্থার কথা আলোচনা করেছিলাম, সেই বিমূর্ত একটা চিহ্ন-কাঠামো, যা দিয়ে আমি কম্পিউটারকে বুঝছি। এই যে হার্ড ডিস্ক বুঝছি /dev/hda বা /dev/hdb দিয়ে, ফ্লপি ডিস্ক বুঝছি /dev/floppy দিয়ে, সিডিরম বুঝছি /dev/cdrom দিয়ে — এগুলো কী? এক একটা নাম, চিহ্ন, এর পিছনে একটা ভৌত বাস্তবতা রয়েছে, যে বাস্তবতাটাকে মাথায় রাখছে কারনেল তথা অপারেটিং সিস্টেম। এই সিম্বলাইজেশন বা চিহ্নীকরণ প্রক্রিয়াটা ঘটতে শুরু করে যেখান থেকে তার নাম ফাইল। ফাইল, তার নিজের নামের আশ্রয়ে বিমূর্ত করে তোলে কিছু বাইট সমাহারকে যাদের ভৌত বাস্তবতাটাকে আমরা তখন আর ভাবছি না, আমার কাছে তখন ফাইল মানেই ডিস্কের গায়ে বরাদ্দ করা ওই বাইট সমাহারে লিখিত তথ্যটা। এতে ওই ঘসঘস করে ঘুরতে থাকা চৌম্বক প্রলেপ লাগানো গোলাকৃতি প্ল্যাটারগুলোকে, তাদের ভীষণ কাছ দিয়ে পরিক্রমণশীল হেডগুলোকে, তাদের নড়াচড়াকে, চৌম্বক তলের উপর ওই হেড দিয়ে তথ্য লেখা বা পড়ার কায়দাকে, ভুলে থাকছি আমি। শূন্য নম্বর দিনে এই নিয়ে আলোচনাটা মনে করুন। যে বাস্তবতাটাকে ফাইল গোপন করে রাখে তার নামের পিছনে। এখন থেকে ওই বাইট সমাহার পরিচিত হবে ওই নাম দিয়ে। মাথায় রাখুন, তথ্য মানে তথ্যহীনতাও হতে পারে। ব্ল্যাংক বা ফাঁকা ফাইল যেখানে কোনো তথ্য লেখা নেই।

প্রথমে এই ফাইল নামকরণের প্রক্রিয়াটাকে বোঝার চেষ্টা করা যাক। ফাইলের নাম শুধু যে আপনি দেন তা নয়, আপনি যে প্রোগ্রামগুলো চালাবেন তারা বহু ফাইল তৈরি করে, অপারেটিং সিস্টেম তার কাজ করার প্রক্রিয়ায় বহু ফাইল বানায়। এই ফাইল বানানো মানেই তো নির্দিষ্ট কিছু ভৌত বাইট সমাহারে গেঁথে রাখা তথ্য সমাহারের জন্য আলাদা একটা পরিচিতি তৈরি করা, আলাদা একটা নাম দিয়ে। আবার অজস্র ফাইল থাকে যারা একটা প্রসেস চলাকালীন তৈরি হয়, প্রসেস মরে গেলে তারাও নিশ্চিহ্ন হয়ে যায়, তাদের আলাদা করে আর সেভ করা হয়না ডিস্কে। প্রসেসটা যতক্ষণ চলে ততক্ষণই তারা বেঁচে থাকে, র‍্যামে বা ভার্চুয়াল মেমরিতে, সোয়াপ ফাইলে, মানে হার্ড ডিস্কে আলাদা করে রাখা অস্থায়ী ফাইল অস্থায়ীভাবে লেখার জায়গায়। এটা খুব ভালো বোঝা যায় কোনো কিছু লেখার সময়। লিখছি, নতুন শব্দগুলো জমা হচ্ছে, জমা হচ্ছে কোথায়? ফাইলটার স্থায়ী শরীরে নয়, কারণ, এখুনি যদি আপনি বেরোতে চান, আপনাকে সিস্টেম জিগেশ করবে, ফাইলটা কি সে সেভ

করবে? তার মানে এখনো তাকে সেভ করা হয়নি। হঠাৎ কারেন্ট চলে গেলে আপনি পরে আর এই না-সেভ করা অংশটা পাবেন না। এই অংশটা অস্থায়ী ভাবে লেখা থাকে, সেভ করার সময়ে তাদের স্থায়ী ফাইলের অংশ করে দেওয়া হয়। হোক অস্থায়ী, তাও তো সেটা একটা লেখা, ওই সময়টুকুর জন্যে তাকে সেইখানে একটা বিশেষ পরিচিতিতে থাকতে হচ্ছে। ধরুন আপনি একই সঙ্গে দুটো ফাইল এডিট করছেন, সেখানে একটার অংশ তো অন্যটায় সেভ হচ্ছেনা, তার মানে ওই অস্থায়ী সময়টুকুর জন্যেই তার একটা পরিচিতি আছে — অস্থায়ী একটা নামও আছে, সিস্টেম যে অস্থায়ী নামের ফাইলে তথ্য লিখছে বা পড়ছে। আপনাকে জানতে হচ্ছে না, কিন্তু নাম একটা আছেই।

২.২।। ফাইলনাম এবং এক্সটেনশন

এই নাম কী ভাবে দেওয়া হবে তার খুঁটিনাটি নিয়ম আবার সিস্টেম থেকে সিস্টেমে আলাদা। ফাইলনামে চিহ্নের সংখ্যা সিস্টেম থেকে সিস্টেমে ৮ থেকে ২৫৫ অব্দি হয়। এবং সংখ্যা বা অক্ষর বা অন্য কিছু চিহ্ন ব্যবহার করা যায়। কোনো কোনো সিস্টেমে বড় হাতের বা ক্যাপিটাল কেস অক্ষর এবং ছোট হাতের বা স্মল কেস অক্ষর আলাদা বলে গণ্য হয়, কোথাও হয়না। যেমন গ্নু-লিনাক্সে dog বা Dog বা dOg বা doG বা DOg বা DoG বা dOG বা DOG প্রত্যেকেই আলাদা ফাইল, কিন্তু এমএসডস বা উইনডোজে এই সব কটা নামই চিহ্নিত করবে একই ফাইলকে, সেটা সে দেখাবে dog বা Dog বা DOG। আগের আটটার যেকোনো একটা নামে একটা ফাইল থাকলেই সেই ডাইরেক্টরিতে আর ওই আটটার অন্য কোনো নামে কোনো ফাইল তৈরি করতে দেবেনা। এমএসডস বা উইনডোজ ফাইলনামে দুটো অংশ থাকে। কাঠামোটা হল *.*। এই কাঠামোয় প্রথম * টা এমএসডসে হত ৮ বা তার কম অক্ষরের, পরের * টা তিন বা তার কম। প্রথম *-টাকে বলে ফাইলনেম, আর পরের *-কে বলে এক্সটেনশন। উইনডোজে একই সঙ্গে প্রতিটা ফাইলনামের দুটো সংস্করণ লিখে রাখা হয় — একটা ওই এমএসডসের ছক মেনে, ৮.৩-এ, অন্যটা হল ২৫৫ চিহ্নের, ওই ছকটা না-মেনে। দুটোই ব্যবহার করা যায়। এমএসডস প্রম্পটে যেমন দেখবেন, দুটো নামই দেখায়, প্রাথমিক ভাবে ডস নামটা, ডানদিকে উইনডোজ নামটা। আপনি যদি উইনডোজে C:\ মানে রুট ডিরেক্টরিতে দাঁড়িয়ে cd My Documents লিখে এন্টার মারেন, ডস প্রম্পট আপনাকে দেখাবে — Too many parameters – Documents, অর্থাৎ সিস্টেম এই নামটার মানেই বুঝতে পারছে না। কিন্তু cd mydocu~1 লিখে এন্টার মারলে পৌঁছে দেবে C:\My Documents\ ডিরেক্টরিতে। কিন্তু উইন্ডোজ এক্সপ্লোরার দিয়ে সিস্টেম ব্রাউজ করার সময়ে দেখাবে গোটা নামটাই। এটা তার দেখানোর নাম, সিস্টেম তাকে বোঝে 'mydocu~1' দিয়ে, যার অক্ষরের সংখ্যা ৮, সেই ডস নিয়ম।

গ্নু-লিনাক্সে সিস্টেমের কাছে এই এক্সটেনশন ব্যাপারটার কোনো মানেই নেই, মানেটা আছে আপনার কাছে, আপনি একই ধরনের অনেকগুলো ফাইলকে একই এক্সটেনশন দিয়ে রাখলে পরে আপনার চিনতে সুবিধে হবে, যে কারণে আমরা প্রবন্ধগুলোর নাম দিচ্ছিলাম essay.1.text, essay.2.text, essay.3.text, ... ইত্যাদি। এখানে দেখুন নামের আসলে দুটো নয় তিনটে খণ্ড। এই নামগুলো সংখ্যা ব্যবহার করেছে, সেখানে শুধু অক্ষর ব্যবহার করা যেতে পারত। আমি যেমন এই নামগুলো দিতাম, 01.essay.text, 02.essay.text, 03.essay.text, ... ইত্যাদি, যাতে আমায় ব্যাশ কমান্ড কমপ্লিশনের কাছে শুধু গোড়ার দুটো অঙ্ক দিয়ে ট্যাব মারলেই কাজ চলে যায়। আর প্রথম ডট বা বিন্দুটা দেওয়ার একটা কারণ এই যে ls মারলে পরপর দেখতে বেশ সুন্দর লাগে। একদম ডানদিকের যে অংশটা '.text' — এর থেকে আপনি, সিস্টেম নয়, আপনি, চেনেন যে এটা একটা টেক্সট ফাইল। আপনি এটা '.txt' দিতে পারতেন, '.essay' দিতে পারতেন, '.probondho' দিতে পারতেন, বা যা আপনার খুশি। এর আবার কিছু প্রথা বা কনভেনশনও গড়ে উঠেছে। যেমন, '*.c' মানে সি প্রোগ্রামের কোড লেখা ফাইল, '*.o' মানে কম্পাইলেশনের সময়ে তৈরি অবজেক্ট ফাইল, '*.html' মানে নেটের হাইপার টেক্সট মার্কআপ ল্যাংগোয়েজের ফাইল, '*.ps' মানে পোস্টস্ক্রিপ্ট ফাইল, '*.mp3' মানে এমপেগথ্রি মানে যার থেকে গান শুনি, '*.avi' মানে যার থেকে সিনেমা দেখি ইত্যাদি। বলুন তো '*.h' বা '*.tar.bz2' বা '*.pdf' কী ধরনের ফাইল? মনে করার চেষ্টা করুন, প্রত্যেকটাই উল্লেখ করা আছে আগের দিনগুলোর লেখায়। আর আন্দাজ করার চেষ্টা করুন তো, '*.conf' কী ধরনের ফাইল? একটা জিনিষ দেখুন, এই যে আমরা এক্সটেনশনগুলোকে উল্লেখ করছি '*.' দিয়ে, এর মানে ওই '*' অংশে যা খুশি আসতে পারে, তার পর একটা বিন্দু বা ডট '.' থাকবে, তারপর আসবে এক্সটেনশনটা। এটা একটা রেগুলার এক্সপ্রেশন, মানে একটা ছক, শেল যা দিয়ে সব গুলো প্রবন্ধের ফাইলকে চিনেছিল পাঁচ নম্বর দিনে।

এবং এই যে বললাম না, 'যা খুশি আসতে পারে' — এটা যে কী ভীষণ বিপজ্জনক, তা টের পাওয়ার জন্যে আপনি হাতে নাতে একটা জিনিষ করতে পারেন। একটা ফাইল তৈরি করুন। ফাইল তৈরি করার দুটো খুব সহজ উপায় হল টাচ এবং ক্যাট। টাচ আসলে অন্য কাজের জন্যে তৈরি, তার একটা ব্যবহার এই ফাইল বানানো। ধরুন কমান্ড দিলেন '`touch essay`'। এর মানে '`essay`' নামে একটা ফাইল তৈরি হয়ে যাবে আপনি যে ডাইরেক্টরিতে আছেন সেখানে। কিন্তু এটা হবে শূন্য বা ব্ল্যাংক ফাইল। এত শূন্য বা এত ব্ল্যাংক যে তাদের মধ্যে খড়ও ভরা নেই, বা ভাঙা কাঁচের উপর ইঁদুরের পায়ের শব্দের মত শুকনো গলায় তারা ফিসফিসও করেনা, শুধু একটা পরিচিতি — একটা ফাইলনাম, এবং তার সঙ্গে একটা টাইমস্ট্যাম্প, মানে কখন বানানো হল, আর আরো দু-চারটে তথ্য।

ওই একই ফাইল ক্যাট দিয়ে বানাতে হলে আমাদের ব্যবহার করতে হত '`cat > essay`'। এখন আমরা আর এন্টার মারার কথা উল্লেখ করছিনা, এত বার করেছি, যে সেই ব্যাপারটা আমাদের মাথায় এন্টার করে গেছে বলেই আশা করা যায়। ক্যাট দিয়ে কাজ করতে চাইলে কাজটা কিন্তু এখানেই শেষ নয়। (ক্যাটের কাজটা কী বলুন তো, আগেই বলে এসেছি, মনে করতে পারছেন?) ক্যাট এর চিহ্নটা দেখুন, ও রিডাইরেক্ট করছে, চালান করছে। কিন্তু কাকে? আপনাকে, মানে আপনার টাইপকে, এখন আপনি যা টাইপ করবেন। যা খুশি। দেখবেন, আপনার টাইপ করার জায়গা বানাতে ক্যাট কমান্ড প্রম্পটটাকেই হাওয়া করে দিয়েছে, সেখানে জ্বলজ্বল করছে কার্সর, আপনার টাইপ করা অক্ষর যেখানে ফুটে উঠবে।প্রতি লাইনের শেষে একবার করে এন্টার মারবেন, নতুন লাইন শুরু হবে, তারপরে যখন আপনার মনে হবে, টাইপ করতে করতে আপনি নিজেই প্রায় টাইপরাইটার হয়ে গেছেন, তখন আর কিছু এন্টার করবেন না, একজিট করবেন, কন্ট্রোল-ডি মেরে, দেখবেন এতক্ষণের অদৃশ্য কমান্ড প্রম্পট ফের সগৌরবে ফিরে এসেছে। এবং একবার `ls` মারলে দেখতে পাবেন আপনার বানানো ফাইল। আর তার মধ্যে আপনার যে সূক্তিরত্নাবলী টাইপ করে রেখেছিলেন সেটা দেখতে চান, মারুন '`cat essay`'। স্ক্রিনে দেখিয়ে দেবে। এবার আর এটা শূন্য নয়, আপনার টাইপ করা লেখা ভরা রয়েছে।

এবার, খ্যাঁচটা এইখান থেকে শুরু, নামটা '`essay`' না-দিয়ে দিন '`-essay`'। মজার কথা কী বলুন তো এই নামে ফাইল আপনি '`touch -essay`' দিয়ে তৈরিই করতে পারবেন না। যদিও, '`cat > -essay`' দিয়ে পারবেন। 'ls' সেভাবেই দেখাবে এটাকে যে ভাবে অন্য ফাইলদের দেখায়, আপনি যে নাম দিয়েছেন সেই নামেই। তাহলে? সমস্যাটা কোথায়? এবার, আপনি '`rm -essay`' দিয়ে এটাকে ওড়ানোর, '`mv -essay`' দিয়ে নাম বদলানোর, বা '`cp -essay essay`' দিয়ে এটাকে '`essay`' নামে কপি করার চেষ্টা করুন তো? এখন অব্দি যতটুকু গ্নু-লিনাক্স জানেন, তা দিয়ে চেষ্টা করে করে যদি এটা করে ফেলতে পারেন, তাহলে, এই মুহূর্তে একটা এপিঠ-ওপিঠ করান, নিজের নাম বদলে ফেলুন আইনস্টাইনে। কেন? কারণ, একটু আগেই যেমন আমরা দেখলাম, আগে হাইফেনটা থাকলেই সেটাকে অপশান বলে মনে করে নিচ্ছে touch বা rm বা mv বা cp। নিজেই চেষ্টা করে দেখুন না।

অনাইস্টাইনদের জন্যে পরিজ্ঞাতব্য তথ্য এটাই যে, এই ফাইলটাকে সাইজ করার কায়দা আমি এই লেখায় বলে দেবনা, ইচ্ছে করেই, যাতে গ্নু-লিনাক্স ডকুমেন্টেশন ঘাঁটার অভ্যেশটা হয় আপনার। যা গ্নু-লিনাক্স শেখার একটা প্রাথমিকতম শর্ত। যদি নিজে নেট থেকে টেক্সট নামিয়ে নেন বা বই কিনে নেন, তাহলে তো ভালোই। নইলে জিএলটির সিডিটার জন্যে মেল করবেন, একশো টাকা করে নিচ্ছি আমরা, ডাউনলোড আর সিডি পোড়ানোর খরচ বাবদ, তাতে গ্নু-লিনাক্স সংক্রান্ত ছহাজারের উপর ওয়েব পেজ এবং ছশোর মত পিডিএফ ফাইল রাখা আছে, যার গোটাটা শেখা হয়ে গেলে আপনি অনায়াসে আন্তর্জাতিক খেতাব পরমলিনাক্সচক্রের জন্যে দরখাস্ত করতে পারেন, ওটা পেয়ে গেলে বিনা পয়সায় ইন্টারনেট করা যায়, আর সারা বছরই ঠ্যাং ভাঙার ছুটি দেয়। যারা দূরে থাকেন, লিখবেন, জিএলটি-র সিডি পোস্টে পাঠিয়ে দেব।

এবার একটা জিনিষ মাথায় রাখুন, যারা উইনডোজে কাজ করে অভ্যস্ত, তারা জানেন, উইনডোজ এক্সপ্লোরারে ঘুরে বেড়াতে বেড়াতে দেখা যায়, এক একটা বিশেষ ধরনের ফাইলের পাশে এক একটা বিশেষ ধরনের ছবি বা আইকন। এদের যে কোনোটায় মাউস দিয়ে ক্লিক করলেই একটা বিশেষ প্রোগ্রাম তাদের খোলে। যেমন যেসব ফাইল '*.doc', তাদের যেকোনোটায় ক্লিক করলেই সেটাকে খুলবে এমএস-ওয়ার্ড, '*.xls' হলে তাদের খুলবে এমএস-একসেল, '*.txt' হলে খুলবে নোটপ্যাড, বা খুব বড় সাইজ হলে, উইন্ডোজ ওয়ার্ডপ্যাড। এদের সিস্টেম চিনছে কিন্তু ফাইল এক্সটেনশন দিয়েই, মানে ডস ফাইলনাম কাঠামোর ওই ৮.৩-এর পরের তিনটে অক্ষর দিয়ে। সিস্টেমের কিরকম বিকট গ্যাঁড়ামো দেখুন, আপনি যখন ৪ অক্ষরের কোনো এক্সটেনশন ব্যবহার করছেন, 'glinux.html', তখনো সিস্টেম তাকে চিনছে ওই ৮.৩-এই, আপনি এমএসডস প্রম্পটে গিয়ে ডির (dir) মেরে দেখুন, সিস্টেম দেখাচ্ছে 'glinux~1.htm'। এই নামটা দিয়ে টাইপ কমান্ড মেরে দেখুন, '`type glinux~1.htm`', দিব্য দেখবেন সোনামুখ করে ওয়েবপেজটার এইচটিএমএল কোডিং দেখিয়ে দিল। আবার আপনি উইনডোজ এক্সপ্লোরারে যান, দেখবেন ওই ফাইলটার পাশে ওয়েবপেজের আইকন ফুটে উঠেছে, এবং ক্লিক করলে তাকে ইন্টারনেট এক্সপ্লোরার দিয়েই খুলছে। এই কাজের নাম আর দেখানোর নামের কথা তো আমরা আগেই বলেছি। এবার মজাটা হল, আপনি কোনো একটা নামের এক্সটেনশন মানে পরের তিন অক্ষরের অংশটা বদলাতে যান, দেখবেন কী চেঁচায়, 'তুমি এসব করলে যদি আর কখনো ফাইলটা খুলতে না-পারো, আমি কিন্তু কিছু জানিনা', আসলে জানেনা ও নিজেই। এক্সটেনশন বদলে ক্লিক করে দেখুন, ফাইলটা ও খুলতেই পারবে না। এবং প্রত্যেকটা ব্যবহারকারীকেই নিজের মত নির্বুদ্ধি বলে ধরে নেয় উইনডোজ। গ্নু-লিনাক্স-এর মত নিরাপত্তা ব্যবস্থা না-থাকায় আমি আগে যা করে রাখতাম সেটা হল কোনো '*.doc' বা '*.txt' ফাইলের এক্সটেনশনটা বদলে কোনো সিস্টেম ফাইল এক্সটেনশন দিয়ে রাখা যেমন '*.drv' বা '*.sys' ইত্যাদি। কেউ যদি ফাইলটা পায়, এবং ওরকমই পাঁঠা হয়, খুলতেই পারবে না। এবং সিস্টেমের গ্যাঁড়ামো এখানেই শেষ নয়, নিজের এই বোকামিটা আবার গোপন করারও চেষ্টা করে উইনডোজ। ডিফল্ট সেটিং-এ করা থাকে ফাইলনাম এক্সটেনশন না-দেখানোর ব্যবস্থা। উইনডোজ এক্সপ্লোরারের 'টুলস' মেনুর 'ফোল্ডার অপশান'-এ গিয়ে তবে অফ করে দিতে হয় 'হাইড ফাইলনেম

এক্সটেনশন'। মানে লোকে দেখবে না, অথচ ও দেখবে, আর এক্সপ্লোরারে পাশে প্রোগ্রামের আইকনটা দেখাবে, যেন ওর কত বুদ্ধি, ম্যাজিক করে জেনে ফেলছে। কোন ফাইলটা কোন প্রোগ্রামের।
এই ম্যাজিকটা সত্যিই 'ম্যাজিক' নামেই সত্যি গ্নু-লিনাক্স-এর বেলায়। এখানে তো কোনো এক্সটেনশন-এর চক্কর নেই, তাহলে কোনো ফাইলকে তার নিজের প্রোগ্রাম কী করে চিনে নেবে এক্স-উইনডোজের ব্রাউজার, বা, আমি নিজে যখন ব্যবহার করতে চাইব কোনো ফাইল, সেটা কী কাঠামোর ফাইল কী করে বুঝব? এতেই ব্যবহার হয় 'ম্যাজিক'। আমরা সে কথায় পরে আসছি।

২.৩।। সাধারণ ফাইলের গঠন

একটা গ্নু-লিনাক্স সিস্টেমে ফাইল হয় নানা রকমের। এর পরের সাবসেকশনে আমরা জানব সেসব। এখানে ফাইলের গঠন বলতে আমরা রেগুলার বা সাধারণ ফাইলের গঠনই বোঝাচ্ছি। কারণ অন্য ফাইলের (ডিভাইস এবং ডিরেক্টরি) গঠন কারনেলের দ্বারা নির্দিষ্ট। রেগুলার ফাইলের গঠনের কিছু তফাতের কথায় আমরা পরে আসছি, যখন আমরা রেগুলার ফাইলদের মধ্যে প্রকারভেদের কথায় আসব। কিন্তু সেই গঠন ফারাকের পরও একটা রেগুলার ফাইলের গঠনের একটা মৌলিক বৈশিষ্ট্য থাকে প্রতি অপারেটিং সিস্টেমেই। সেই গঠনের মধ্যেও কিছু তফাত থাকে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স বা উইনডোজে রেগুলার ফাইলের গঠনে একটা একদেশতা আছে। কিন্তু চিরকাল সব অপারেটিং সিস্টেমের রেগুলার ফাইলের আকার বলতে যে এটাই ছিল তা নয়। এবং এখনো যে রেগুলার ফাইলের অন্য কোনো গঠন হয়না তা নয়। কিন্তু সেগুলো আমাদের আলোচনায় আসবে না। আমরা এখানে গ্নু-লিনাক্সের রেগুলার ফাইলে বাইট রাখার কায়দা নিয়ে দু-একটা কথা বলে নেব।
যুগ থেকে যুগে, অপারেটিং সিস্টেম থেকে অপারেটিং সিস্টেমে আম রেগুলার ফাইলের গঠনের কায়দা এক নয়, একাধিক। তার মধ্যে একটা কায়দা হল যেখানে একটা ফাইল মানেই পরপর কিছু বাইটের সমষ্টি। যে বাইটগুলো আলাদা করে খুব ব্যক্তিত্ববান চরিত্রবান ইত্যাদি হতেই পারে, কিন্তু তাদের ব্যক্তিত্ব বা চরিত্র বিষয়ে অপারেটিং সিস্টেম কোনো খোঁজই রাখে না, শত চাইলেও কোনো ক্যারেকটার সার্টিফিকেট দিতে পারবে না। অপারেটিং সিস্টেমের কিছু এসে যায়না। অপারেটিং সিস্টেম শুধু একটা খোঁজই রাখে, তাদের বাইটবানতা। মানে, তার ভাণ্ডারে কত বাইট আছে, কোন বাইট, কোন এলাকার কত নম্বর বাইট, ইত্যাদি। বাইটগুলো অর্থময় হয়ে ওঠে ব্যবহারকারীর স্তরে, বাইটগুলো থেকে অর্থ তৈরি হয় ইউজার লেভেল প্রোগ্রামগুলোর হাতে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স, এবং উইনডোজও, ফাইলগুলোকে এভাবেই দেখে। অপারেটিং সিস্টেমের কাছে ফাইলগুলো বাইট-সমাহারের চেয়ে বেশি কিছু না-হওয়াটা কাজের পক্ষে সবচেয়ে সহজ এবং নমনীয় একটা অবস্থা তৈরি করে। ইউজার লেভেল প্রোগ্রামগুলো তাদের যেভাবে প্রয়োজন যা প্রয়োজন তাই রাখতে পারে ফাইলে, নিজেদের কাছে সবচেয়ে সুবিধেজনক রকমে যা খুশি নাম দিতে পারে ফাইলের। অপারেটিং সিস্টেম যেমন কোনো সাহায্য করেনা তেমনি কোনো বাধাও দেয়না। আর কোনো ঝুঁকিপ্রবণ ব্যবহারকারী যখন তার নিজের রকমে অপারেটিং সিস্টেম নিয়ে পরীক্ষা-নিরীক্ষা বা খেলাধুলো করতে চায়, সেখানে এটা একটা জরুরি ব্যাপার হয়ে উঠতে পারে।
এছাড়া অন্য যে আকারের ফাইল হত বা আজো কোনো বিশেষ বিশেষ কাজের জন্যে হয়, তাদের আমরা আর আলোচনায় আনছি না। কৌতূহল মেটানোর জন্যে বলে রাখি, এক ধরনের ফাইলে তথ্য থাকত বিশেষ একটা নথি বা রেকর্ডের ছকে। যেমন আশি-কলাম পাঞ্চড কার্ডের যুগে, অনেক মেইনফ্রেম অপারেটিং সিস্টেমেই ব্যবহার হত এই ফাইল। এই ফাইলে তথ্য লেখা মানে আর একটা রেকর্ড যোগ করা, যাতে আছে আশিটা চিহ্ন বা ক্যারেকটার, কিম্বা পুরোনো একটা রেকর্ড উড়িয়ে সেখানে এই নতুন রেকর্ডটা গুঁজে দেওয়া। বা ১৩২ কলাম লাইন প্রিন্টারের জন্যে রেকর্ডের সাইজ হত ১৩২ টা ক্যারেকটারের। আবার এখান থেকে তথ্য পড়া মানে একটা একটা করে রেকর্ড পড়া। এখন আর এই ফাইলের ব্যবহার হয়না। কিন্তু অন্য আর এক রকমের ফাইল আছে যেখানে তথ্যটা থাকে গ্নু-লিনাক্স ফাইলের মত অসংগঠিত বিট সমাহারের আকারে নয়, সেখানে তথ্যের সবসময়ই একটা কাঠামো থাকে, একটা ট্রি বা ডালপালাসহ গাছের মত করে সাজানো থাকে তথ্যটা। এই ধরনের ফাইল এখনো ব্যবহার হয় কোনো কোনো বাণিজ্যিক ডেটাবেস নিয়ে কাজ করা মেশিনে। এই ধরনের ফাইলে তথ্য থাকে নির্দিষ্ট কিছু কি বা বিশিষ্ট বিন্দুকে কেন্দ্র করে, এবং যখন সেই বিশিষ্ট বিন্দুর বিষয়ে কোনো তথ্য খোঁজা হয়, তখন জানার দরকার পড়েনা ফাইলের কোন জায়গায় তথ্যটা আছে, ওই বিশিষ্ট বিন্দুর সাপেক্ষে সেটা অপারেটিং সিস্টেমই বার করে দেয়। অর্থাৎ এখানে ফাইলে তথ্য রাখা এবং পড়ার ব্যাপারে অপারেটিং সিস্টেম সরাসরি অংশগ্রহণ করে। কিন্তু এই দু ধরনের কোনো ফাইলই আমাদের প্রয়োজন পড়বে না, আমাদের কাছে ফাইল মানেই কিছু বাইটের সমষ্টি, যাদের বাইট কাঠামো ছাড়া আর কিছু অপারেটিং সিস্টেম জানেনা। সেসব জানে সেই প্রোগ্রাম যে ফাইল নিয়ে কাজ করে।
কিন্তু তাই যদি হয়, তাহলে, অপারেটিং সিস্টেম যদি নাই জানে কোন ফাইলে কী আছে, তাহলে এক নম্বর দিনে আমরা যে বিভিন্ন ধরনের ফাইলের পার্থক্যের কথা বললাম, কোনো ফাইল টেক্সটের, কোনো ফাইল কোডের, কোনো ফাইল প্রোগ্রামের, যাদের নাম লিখে এন্টার মারলেই চলতে শুরু করে, সেই তফাতটা ঘটছে কী করে?

২.৪।। রেগুলার ফাইল

একটা গ্নু-লিনাক্স সিস্টেমে তিন ধরনের ফাইল থাকে, রেগুলার বা সাধারণ ফাইল, ডিরেক্টরি ফাইল, এবং ডিভাইস ফাইল। এইমাত্র ফাইলের তথ্যের কাঠামো নিয়ে আমরা যে আলোচনাটা করলাম সেটা এই রেগুলার বা অর্ডিনারি বা সাধারণ ফাইল। কারণ, ডিরেক্টরি ফাইল আর ডিভাইস ফাইলের চরিত্র একদম ভিন্ন ধাঁচের। তার কথায় আমরা পরে আসছি। রেগুলার বা সাধারণ ফাইলে থাকে বিভিন্ন ধরনের ব্যবহারকারীর স্তরের তথ্য। এমনকি যখন সিস্টেমের কোনো প্রসেস এই রকম কোনো ফাইল লেখে, সেই মুহূর্তে সেই প্রসেসও একজন ব্যবহারকারী। যেমন পরে আমরা যখন ইউজার আইডি বা ব্যবহারকারীর পরিচিতি কাকে বলে জানব, দেখব, একটা গ্নু-লিনাক্স অপারেটিং সিস্টেমে ব্যবহারকারীদের তালিকায় নাম থাকে man, games, mail, news, postfix ইত্যাদি একাধিক সিস্টেম প্রসেসের। পাঁচ নম্বর দিনের আলোচনা থেকে /etc/passwd ফাইলটা মিলিয়ে দেখুন। যে কোনো ব্যবহারকারী না-হলেও, সর্বোচ্চ ব্যবহারকারী মানে রুট যে কোনো সময়েই এর যে কোনো ফাইলে কী আছে দেখতে পারে। অথচ কোনো ডিরেক্টরি ফাইল এমনকি সে-ও পড়তে পারেনা। ডিরেক্টরির মধ্যে কী কী ফাইল বা সাবডিরেক্টরি আছে সেই তালিকাটা দেখতে পারে, কিন্তু খোদ ওই ডিরেক্টরি ফাইলটা পড়তে পায়না। যদিও সে যে কোনো সময় যে কোনো ডিরেক্টরি ওড়াতে পারে বা নাম বদলাতে পারে। ডিরেক্টরি ফাইলরা হল সিস্টেম ফাইল, যারা সিস্টেমের কাঠামো ধরে রাখে।

রেগুলার বা সাধারণ ফাইলগুলোয় সচরাচর ভরা থাকে অ্যাসকি টেক্সট বা বাইনারি তথ্য। সিস্টেমের কিছু এসে যায়না, ফাইলে বাইনারি আছে না অ্যাস্কি আছে না গুষ্টির পিন্ডি আছে, এসে যায় ব্যবহারকারীর। সিস্টেম এই বাইনারি বা অ্যাসকির ভিতর কোনো তফাত করেনা, সবই তার কাছে রেগুলার ফাইল। শূন্য নম্বর দিনের আলোচনাগুলো মনে করুন, এই তথ্যের কাঠামো নিয়ে আমরা যেখানে কথা বলেছি। আর এক নম্বর দিনে আমরা বলে এসেছি বাইনারি, অ্যাস্কি ইত্যাদি বিভিন্ন ফাইলের কাজ কী? মনে করতে পারছেন? আগেই বলেছি, অ্যাসকি ফাইলে থাকে লাইন লাইন টেক্সট। এর সুবিধেটা এই যে খুব সহজে এদের স্ক্রিনে দেখা যায়, প্রিন্ট নেওয়া যায়, কোনো এডিটর দিয়ে এডিট করা যায়।

আর একবার মনে করিয়ে দিই — ওয়ার্ড প্রসেসর নয়, এডিটর। ওয়ার্ড প্রসেসর মানে ওয়ার্ডকে প্রসেস করে, প্রসেসিত ওয়ার্ডের নানা চিহ্ন গুঁজে দেয় নির্জলা, র, টেক্সট-এর মধ্যে, সিস্টেম তখন আর তাকে অ্যাসকি টেক্সট হিশেবে কাজে লাগাতে পারেনা। ধরুন, লিলো-কনফ ফাইলটা, এর প্রথম লাইনটাকে আপনি বোল্ড করেছেন আপনার ওয়ার্ড প্রসেসরে, প্যারাগ্রাফগুলোকে বাঁদিক থেকে আধ ইঞ্চি সরিয়ে এনে ইনডেন্ট করেছেন, এবার সেই ফাইলটায় আপনার ওয়ার্ড প্রসেসর এই সব করার আদেশগুলো নানা বিচিত্র রকমে এবং চিহ্নে গুঁজে দিল। এবার সিস্টেম যখন, বুট করার সময়ে, লিলো-কনফ পড়তে গেল, কোথায় কোন কারনেল আছে জানার জন্যে, সে ফাইলের মধ্যে গিয়ে পেল সেইসব দেবশিশুদের পদচিহ্ন, কিন্তু সিস্টেম তার বাপের জন্মে, ইউনিক্স মাল্টিক্স এফএমএস, কোনোদিন এসবের মানে জানেনা, পড়তে পারল না, এবং ভারি হিংসুটে সে, এটা হল কর্নিশ অগরের কাছে যাওয়ার আগের ফেজ, সিস্টেমের দৈত্য কিছুতেই বুট করতে রাজি হলনা, স্ক্রিনে ফুটে উঠল 'বুট ফেইলিওর . . . কারনেল প্যানিক'। প্যানিক শুধু কারনেলের, আপনার প্রাণও হঁক হঁক করছে। ভাবছেন, কী দরকার ছিল এসব অক্ষরের শব্দের প্যারার সৌন্দর্যবোধ ওই কেলে চাঁড়াল দৈত্যকে শেখাতে যাওয়ার, একি কালচারাল পাবলিক নাকি, সৌন্দর্য এবং বোধে গলে গলে পড়ছে। এর জন্যে দরকার কোনো ম্যাটার-অফ-ফ্যাক্ট এডিটর, যে নিখাদ টেক্সট আর লাইন-ভাঙার চিহ্ন ছাড়া আর কিছু গুঁজবে না ফাইলের মধ্যে, যেমন ইম্যাক্স, ভিম, জো, জেডিট, গেডিট, নেডিট, কেডিট ইত্যাদি। ওপেন অফিস রাইটার, অ্যাবি-ওয়ার্ড, কে-অফিস ইত্যাদি ওয়ার্ড প্রসেসর দিয়ে হবেনা।

এই অ্যাস্কি ফাইলগুলোকে শুধু যে এডিট করা যায় তাই নয়, যদি অনেক প্রোগ্রামই ইনপুট আউটপুট হিশেবে এই অ্যাস্কি ফাইল ব্যবহার করে, তখন এদের একটার আউটপুটকে অন্যটার ইনপুটে জুতে দেওয়া যায়, শেলের পাইপলাইন দিয়ে, পথনির্দেশ করে। একটু আগে দেখেননি, '`man -k CD | grep 'audio'> audio.cd.program.text`' কমান্ডে? ম্যান প্রথমে সমস্ত ম্যানুয়াল পাতা ঘেঁটে দেখল, কোথায় কোথায় সে পাচ্ছে '`CD`' শব্দটা। এই '`man -k CD`' কমান্ডের আউটপুট মানে সেই ৩৮ লাইন অ্যাস্কি আউটপুট চলে গেল গ্রেপ কমান্ডের কাছে ইনপুট হয়ে। '`grep 'audio'`'। গ্রেপ এবার তার সামনে পেল ওই আটত্রিশ লাইন ইনপুট, তাদের মধ্যে সে খুঁজে দেখল কোনো লাইনে '`audio`' শব্দটা আছে কিনা। এবার গ্রেপ কমান্ডের যে ছ লাইন অ্যাস্কি আউটপুট, যেটা আগে আমরা দেখছিলাম স্ক্রিনে, সেটা রিডাইরেক্ট বা চালান হয়ে গেল একটা অ্যাস্কি টেক্সট ফাইলে, যার নাম '`audio.cd.program.text`'।

সাধারণ বা রেগুলার ফাইলদের মধ্যে সবচেয়ে সংখ্যাগরিষ্ঠ এই অ্যাস্কি টেক্সট ফাইল। শূন্য নম্বর দিনে আমরা যে চিহ্ন বা ক্যারেকটারের কথা বলেছিলাম, যাদের স্ক্রিনে দেখা যায়, বা ছাপা যায়, তাদের দিয়েই ভরা থাকে এই ফাইলগুলো। ls, who, tar, less, wc, man, ইত্যাদি যে কমান্ডগুলো আমরা ব্যবহার করছিলাম তারা কিন্তু এই অ্যাস্কি ফাইল নয়, বা সি কোড লেখা কোনো ফাইল আপনি কম্পাইল করার পর যে একজিকিউটেবল বা চালানো-যায়-এমন বাইনারি ফাইল পাবেন, তারাও অ্যাস্কি টেক্সট ফাইল নয়। টেক্সট ফাইলের গোটা টেক্সটা ছোট ছোট দলে ভাঙা থাকে, এক একটা লাইনে এক এক দল চিহ্ন। প্রতিটি

লাইন শেষ হয় 'লাইনফিড' নামের চিহ্নে, অ্যাসকি সিস্টেমে যার ডেসিমাল বা দশমিক মান ১০, মনে করতে পারছেন? এই 'লাইনফিড'-কে 'নিউলাইন' বলেও ডাকা হয়। এই লাইনফিড বা এলএফ চিহ্নটার বিশেষত্বটা দেখুন, এই চিহ্নটাকে কিন্তু আপনি দেখতে পাননা, না স্ক্রিনে না প্রিন্টে, দেখতে পান তার কাজের নমুনাটাকে। যেই একটা লাইন ভেঙে আর একটা নতুন লাইন শুরু হয়। টেক্সট লেখার সময় আপনি যেই এন্টার মারেন, অমনি সেখানে একটা লাইনফিড ঢুকে পড়ে। পরিচিত ওয়ার্ড-প্রসেসরগুলোয়, যখন আপনি অপশানে দেন 'শো নন-প্রিন্টিং ক্যারেকটারস', অদৃশ্য চিহ্নগুলো দেখাও, তখন অন্য অনেক বাড়তি চিহ্নের সঙ্গে এই লাইনফিড-কেও দেখায়, সচরাচর এর চিহ্নটা থাকে '¶'। কিন্তু একটা জিনিষ খেয়াল রাখবেন, ওয়ার্ড প্রসেসর এটাকে দেখায় শুধু আপনি যেখানে লাইন ভেঙেছেন, মানে এন্টার মেরেছেন, স্ক্রিনের বা প্রিন্ট-এর পাতার মাপের সঙ্গে মিলিয়ে মিলিয়ে লাইনকে সে ভেঙে দেয় নিজেই, যার টেকনিকাল নাম ওয়ার্ড-র‍্যাপ, যা একটা বিশুদ্ধ টেক্সট এডিটর করেনা, যদি আপনি আলাদা করে না-বলে দেন, এমনিতে সেখানে লাইন ভাঙে শুধু যখন আপনি ভেঙেছেন, বা লাইনফিড দিয়েছেন এন্টার মেরে।

আর অন্য ফাইল হল বাইনারি ফাইল, মানে, এক কথায় না-অ্যাস্কি রেগুলার ফাইল। স্ক্রিনে ক্যাট করলে বা প্রিন্ট নিলে ঠিক সেই জাতের জঞ্জাল তৈরি হয়, যার সাথে আমরা পরিচিত হয়েছিলাম, শূন্য নম্বর দিনে, ভিশুয়াল চ্যাংড়ামোয়। আমাদের কাছে মনে হয় বিশুদ্ধ এবং পরিণামহীন পরিচ্ছন্ন জঞ্জাল, কিন্তু 'ls' খুলে দেখেছি, প্রত্যেকবার আমরা যখন এলএস মারি, শেল তাকে খুঁজে বার করে '/bin/ls' থেকে, সিস্টেম তাকে পড়ে ফেলে এবং এক্সিকিউট করে, কী বস্তু সে প্রতিবার পড়ে, এই কৌতূহল থেকে, সেই ফাইলে একদম প্রথম দিকেই অন্তত দুটো চেনা শব্দ খুঁজে পেয়েছিলাম, 'gnu' আর '/lib/ld-linux.so.2'। দ্বিতীয়টা ঠিক কী ধরনের চীজ কোনো আন্দাজ করতে পারছেন? কোনো আলগা আন্দাজ? আমাদের কাছে যেটা এলোপাথাড়ি জঞ্জাল বলে মনে হচ্ছে, অবভিয়াসলি তার মধ্যেও একটা কাঠামো আছে, মেশিন ভাষার কাঠামো। যা থেকে প্রোগ্রামটা কাজ করে। যদিও গোটাটাই হল পরপর কিছু বাইটের সমাহার, কিন্তু এর মধ্যেই একটা কিছু থাকে যা থেকে কোনো ফাইলকে সিস্টেম একজিকিউট করে, চালায়, অন্য প্রোগ্রামকে চালায় না, তার মানে সিস্টেম একে একজিকিউটেবল বলে চিনতে পারে।

একটা একজিকিউটেবল বাইনারি ফাইলের মূলত পাঁচটা খণ্ড থাকে — হেডার, টেক্সট, ডেটা, রিলোকেশন বিট, এবং সিম্বল টেবল। আর এই হেডার শুরু হয় সেই ম্যাজিক নাম্বার দিয়ে, যা থেকে সিস্টেম এই ফাইলটাকে একটা একজিকিউটেবল ফাইল বলে চিনতে পারে। এই ম্যাজিক নাম্বার দিয়ে বিভিন্ন ধরনের ফাইলকে চেনে সিস্টেম। গ্নু-লিনাক্সে একটা কমান্ড আছে, ফাইল (file), যা ফাইলের মধ্যেকার হেডারের ওই ম্যাজিক নাম্বার আর অন্য প্রয়োজনীয় তথ্য পড়ে আপনাকে জানিয়ে দেয় ফাইলটার সম্পর্কে আপনি যা জানতে চান। সিস্টেমও এই ভাবেই জানে, এর জন্যে কোনো ফাইল এক্সটেনশন জাতীয় কায়দা বা তাকে জানানোর জন্যে আরো আরো কায়দা তার দরকার পড়েনা। 'file /bin/ls' আর 'file essay.1.text' করে দেখুন তো কী দেখাচ্ছে। বা, যে কোনো ফাইলকে। ফাইলের কোনো ব্যবহারগত বৈশিষ্ট্য বা অ্যাট্রিবিউট, যেমন কার মালিকানা এই ফাইলের উপর, কে কে একে পড়তে বদলাতে বা চালাতে পারবে, এই জাতীয় কোনো তথ্যই গ্নু-লিনাক্স সিস্টেমে কোনো ফাইলের মধ্যে থাকেনা। এমনকি, হায়, ফাইল নিজে নিজে নিজের নামটা অব্দি জানেনা, সেটাও ফাইলের মধ্যে থাকেনা।

২.৫।। ডিরেক্টরি ফাইল

ডিরেক্টরি ফাইলে কোনো বাইরের তথ্য থাকেনা, শুধু তার মধ্যে যে ফাইল বা সাবডিরেক্টরিগুলো আছে, তাদের কিছু খুঁটিনাটি বিবরণ থাকে। গ্নু-লিনাক্স ফাইলসিস্টেম সাজানো থাকে এরকম ডিরেক্টরির মধ্যে ডিরেক্টরির মধ্যে ডিরেক্টরি দিয়ে। নিজের মালিকানার মধ্যে যেসব ডিরেক্টরি আছে, ঠিক ফাইলের মতো, যখন খুশি ওড়াতে পারে কোনো ব্যবহারকারী, আবার নিজের মালিকানার এলাকায় যেখানে খুশি যেমন খুশি ডিরেক্টরি বানাতে পারে। ডিরেক্টরি ওড়ানোর কমান্ড হল 'rmdir', আর বানানোর কমান্ড 'mkdir'। কিন্তু, একটা ডিরেক্টরিকে 'rmdir' দিয়ে ওড়ানো যায় শুধু তখনই যখন তার পেটে আর কোনো ডিরেক্টরি বা সাবডিরেক্টরি বা ফাইল নেই। সেই অবস্থায় কমান্ড হল '`rm -fr`'। তবে এই কমান্ডটা একটা ভয়ঙ্কর জিনিষ। আপনি রুট হয়ে মানে সুপারইউজার হয়ে, যখন আপনার সব কাজের অনুমতি আছে, কমান্ড দেন '`rm -fr /`', সিস্টেম ফাইল ওড়াতে শুরু করে দেবে। একদম গোড়া থেকে, প্রতিটি ডিরেক্টরির প্রতিটি সাবডিরেক্টরির প্রতিটি ফাইলকে উড়িয়ে যখন থামবে তখন আপনার সিস্টেম হল নিছক একটা শূন্য, হার্ড ডিস্কের আর সার্কিটের ভৌত অবয়ব। নতুন করে বুট করলে সিস্টেম বুটই করবেনা, দেখাবে 'সিস্টেম রিকোয়ার্ড'। আমি সত্যিই করে দেখেছিলাম, সমস্ত প্রয়োজনীয় ফাইলের ব্যাকআপ সিডিতে নিয়ে, আর কোনোদিন কোনো কিছু ফেরত পাওয়া যায়না, এই শূন্য করে দেওয়াটা হয়, টেকনিকাল ভাষায় মিলিটারি ওয়াইপিং। যদি প্রতিটি সফটওয়ার প্রতিটি ডেটাফাইল নতুন করে পয়দা করার উপায় না-থাকে, সাবধান — শূন্য করে ভরে দেওয়া যাহার খেলা তারই সাজে।

ডিরেক্টরি বানাতে হয়, আমরা দেখেছি, যখন কিছু নির্দিষ্ট ধরনের ফাইলকে একই সঙ্গে রাখার দরকার পড়ে। ফাইলগুলোকে সংগঠিত করার দরকার পড়ে। একটা ডিরেক্টরিতে একই নামে একটার বেশি ফাইল থাকতে পারেনা, কিন্তু, আপনি যখন প্রবন্ধের সংকলনের পর সংকলন লিখে চলেছেন, নতুন নতুন ডিরেক্টরি, book.one, book.two, book.three ইত্যাদি, তার

প্রত্যেকটাতেই একটা করে টেবল-অফ-কন্টেন্ট মানে সূচীপত্র মানে toc.text থাকতেই পারে। প্রতিটি ডিরেক্টরিতে, ডিরেক্টরির মধ্যের প্রতিটি ফাইলের জন্যে রাখা থাকে দুটি করে তথ্য রাখার জায়গা বা ফিল্ড। একটা ফিল্ড হল ফাইলটার নাম, অন্যটা ফাইলটার আইনোড নম্বর। একটা ডিরেক্টরিতে যদি দশটা ফাইল থাকে, তার জন্যে থাকবে দশটা এন্ট্রি, প্রতিটিতে ওইরকম দুটো করে ফিল্ড। এই ডিরেক্টরি ফাইলেই লেখা থাকে তার ভিতরকার ফাইলগুলোর নাম, যে নাম আপনি দিয়েছেন, বা কোনো প্রোগ্রাম দিয়েছে। কিন্তু আগেই বলেছি, সরাসরি ডিরেক্টরি ফাইলে লেখার অধিকার কারনেল ছাড়া আর কারোর নেই, এমনকি রুটেরও না। কেউ কোনো ফাইল বা ডিরেক্টরি বদলানো মাত্র সেই ডিরেক্টরির ফাইলে সেই বদলটা লিখে দেয় কারনেল। এইগুলো নিয়ে আরো ভালো করে আলোচনায় আসছি আমরা ফাইল বৈশিষ্ট্য বা অ্যাট্রিবিউট-এর আলোচনায়।

২.৬।। ডিভাইস ফাইল

আমরা দু-নম্বর দিনের ৫ নম্বর সেকশনে ব্লক স্পেশাল এবং ক্যারেকটার স্পেশাল ডিভাইসের কথা বলেছিলাম, সেই দুরকম ডিভাইসের জন্যে গ্নু-লিনাক্সে থাকে দু-রকমের ফাইল — ব্লক স্পেশাল এবং ক্যারেকটার স্পেশাল। আগেই তো বলেছি, গ্নু-লিনাক্সে সবকিছুই এক একটা ফাইল। হার্ড ডিস্ক, প্রিন্টার, সিডি ড্রাইভ, কনসোল, মোডেম, মেমরি — সবকিছুই এক একটা ফাইল। ক্যারেকটার স্পেশাল ফাইল বরাদ্দ থাকে ক্যারেকটার ডিভাইসের জন্যে — অক্ষর বা চিহ্ন বা ক্যারেকটারের ইনপুট এবং আউটপুট করাই যার কাজ, আমি যখন আমার কম্পিউটার থেকে মেল করছি কোনো ইমেল আইডিতে, আমার সিস্টেম কিছু বিশেষ চিহ্ন লিখছে একটা বিশেষ ফাইলে, আউটপুট করছে, যে ফাইলের নাম /dev/modem, মানে আমার চিঠিটা মোডেম বেয়ে নেটে পাঠিয়ে দিচ্ছে। আবার যখন আমি নেট থেকে একটা ওয়েবপেজ পড়তে চাইছি, তখন আবার মেশিন নেট থেকে মোডেমে আসা তথ্য পড়ে ফেলছে ওই একই /dev/modem ফাইল থেকেই, এবার কিছু চিহ্ন বা অক্ষর বা ক্যারেকটারের ইনপুট। ব্লক স্পেশাল ফাইল থাকে হার্ড ডিস্ক ইত্যাদি ব্লক ডিভাইসের জন্যে। নির্দিষ্ট মাপের ব্লক থাকে একক হিশেবে, সেই ব্লকে ব্লকে এখানে তথ্য ইনপুট আউটপুট করা হয়। যখন আমি আমি আমার দ্বিতীয় হার্ড ডিস্কের তিন নম্বর পার্টিশনে কোনো তথ্য রাখছি, বা সেখান থেকে পড়ছি, সেই কাজটা ঘটছে /dev/hdb3 নামের একটা ব্লক ডিভাইস ফাইল দিয়ে। এই ব্লক ডিভাইস এবং ক্যারেকটার ডিভাইস ফাইল এই দুইরকম ফাইলকে মিলিয়ে আমরা এক কথায় বলি ডিভাইস ফাইল। বলুন তো, র‍্যাম বা র‍্যান্ডম অ্যাকসেস মেমরিটা ঠিক কী ধরনের ডিভাইস ফাইল? ব্লক স্পেশাল না ক্যারেকটার স্পেশাল?

এই যে, প্রতিটি ভৌত উপাদানই এক একটা ডিভাইস ফাইল, হার্ড ডিস্ক ড্রাইভ, সিডি ড্রাইভ, ফ্লপি ড্রাইভ, কনসোল বা টার্মিনাল, প্রিন্টার, মোডেম — সমস্ত কিছু — প্রথম প্রথম এটা একটু ঘোলাটে লাগতে পারে, কিন্তু পরে দেখবেন, বেড়ে সুবিধে আছে এর, যেসব কমান্ড দিয়ে কব্জা করা যায় একটা রেগুলার ফাইলকে, সেগুলো ডিভাইস ফাইলের উপরও ব্যবহার করা যায়। যেমন, ক্যাট, কনক্যাটেনেট করে, মানে, স্ক্রিনের ডিসপ্লেতে পেতে দেয় একটা ফাইলকে, বা স্ক্রিন ছাড়া যেখানে চাইবেন, অন্য কোনো ফাইলেও ক্যাট করতে চাইতে পারেন একটা ফাইলকে। আপনি '`cat essay`' কমান্ড দিলে essay ফাইলটাকে স্ক্রিনে পড়ার জন্যে পেতে দেয় ক্যাট। মজার কথা হল, এই একই ভাবে আপনি মাউস ডিভাইস বা /dev/mouse ফাইলটাকেও ক্যাট করতে পারেন। '`cat /dev/mouse`' কমান্ড দিয়ে এন্টার মারুন, এবং দেখুন কমান্ড প্রম্পট ফেরত আসেনি, মানে আপনার আদেশ পালনের কাজটা শেষ হয়নি, চলছে। এবার মাউস নেড়ে দেখুন, প্রফেসর হিজিবিজবিজের কিছু যুগান্তকারী গবেষণা আপনার সামনে ভেসে উঠবে। ক্যাট করতে পারেন সরাসরি আপনার ভৌত হার্ড ডিস্কের একটা ভৌত পার্টিশনকে, করে দেখুন '`cat /dev/hda1`'। খুব মন খারাপ লাগবে, কত বুদ্ধিদীপ্ত প্রবন্ধে ভরে রেখেছেন হার্ড ডিস্ক, তারা আসলে এই? আরো একটা মজার খেলা হয়, কারনেলের কণ্ঠস্বর শোনা। /dev/dsp হল সাউন্ডকার্ড, কথাটা এসেছে ডিজিটাল সিগনাল প্রসেসর থেকে। আর পাঁচ নম্বর দিনে বলেছি, /boot/vmlinuz হল কারনেল। আপনার সিস্টেমের কার্ড কনফিগারেশন যদি ঠিক হয়ে থাকে তাহলে আপনি '`cat /dev/hda1 > /dev/dsp`' কমান্ড দিয়ে কারনেলকে সশব্দ শুনতে পাবেন।

ডিভাইস ফাইলটা এমনিতে কিন্তু ফাঁকা। আপনি তো ফাইলের সাইজ দেখতে জানেন, '`ls -sh /dev/hda1`' বা '`ls -sh /dev/mouse`' মেরে দেখে নিন, দুটোই দেবে শূন্য। অথচ ক্যাট দিলে দেখাচ্ছে। ঠিক এটাই মজা ডিভাইস ফাইলের। ডিভাইস ফাইলরা একদম শেখভের ডার্লিং-এর মত, যখন ভালোবাসে কোনো নাটকের ম্যানেজারকে, তখন তার ভারি দুঃখ, মানুষ কেন ভালো নাটক দেখতে শিখছে না, আবার কোনো কাঠব্যবসায়ীকে বিয়ে করার পর, গির্জে যাওয়ার পথে কিছুতেই রাগ সামলাতে পারেনা, এত যে দেরিতে বর্ষা নামল, ওক কাঠের দর এবার কত পার্সেন্ট বাড়বে বুঝতে পারছো, এদিকে মানুষ শুধু বাজে খরচ করছে থিয়েটার দেখে। ডিভাইস ফাইলের নিজের শরীরে কোনো তথ্যই থাকেনা, সব তথ্যই আসে তার দোসর মানে ভৌত ডিভাইসের কাছ থেকে। তেমনি, নিজে কিছু ধরেও রাখেনা, সবই পাঠিয়ে দেয় ওই ভৌত উপাদানকে। আপনি কোনো ফাইল পাঠালেন '`cp essay /dev/lpt1`' করে, মানে, আপনার 'essay' নামের ফাইলটা '/dev' ডিরেক্টরির 'lpt1' নামের ডিভাইস ফাইলে কপি হতে, ঠিক যেভাবে আপনি একটা রেগুলার ফাইলকে অন্য একটা নামে অন্য কোনো

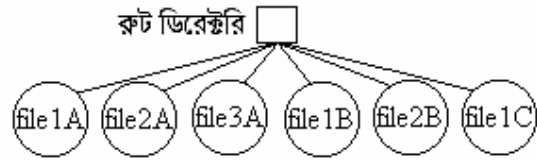
জায়গায় কপি করেন, ওমা, দেখলেন পুরো শ্রমজীবী কোলাহল নিয়ে আপনার পাঠানো ফাইলটা প্রিন্ট হতে শুরু করেছে 'lpt1' নামের প্রিন্টার-পোর্টে লাগানো ডট-ম্যাট্রিক্স প্রিন্টার থেকে। শ্রমজীবী ঘটাং ঘটাং এর জায়গায় ওটা এলিট গুজগুজ ফুসফুস-ও হতে পারে, যদি ডেস্কজেট বা লেজারজেট হয়। এবং আপনার কমান্ড প্রম্পট সঙ্গে সঙ্গেই ফেরত এসেছে, মানে, সিস্টেমের কপি করা হয়ে গেছে, এখন যা ঘটার সব প্রিন্টারে ঘটছে। বা, ফ্লপির ডিভাইস ফাইলে আপনি কপি করছেন মানে ফ্লপিতে রাখছেন ফাইলটা। আপনি ওই কপি (cp) বা মুভ (mv) কমান্ড দিয়েই খালাস, পরের টুকু নিয়ে মাথা ঘামাচ্ছে কারনেল।

২.৭।। এক খাবলা পুনশ্চ

আপাতত আমাদের ফাইল কাকে বলে জানলাম। এবার আমরা যাব ডিরেক্টরি সিস্টেম নিয়ে আলোচনায়, যেখানে আমরা পথনির্দেশের খুঁটিনাটি জানব, কী ভাবে একটা ফাইলের ঠিকানা দেওয়া হয় একটা গ্নু-লিনাক্স সিস্টেমে। তার আগে, দুটো কথা বলে নেওয়ার আছে। ওই '-essay' নামের ফাইলটাকে কিছু করার উপায় না-দেওয়ায় আমার দুজন প্রাথমিক পাঠক খুব চেঁচানোয় উত্তরটা দিয়েই রাখছি। এই ধরনের যে কোনো ফাইলকে কব্জা করার উপায় হল মধ্যে একটা '--' দিয়ে নেওয়া। মধ্যে দেওয়া ওই '--' অংশটা শেলকে বলবে পরের অংশটাকে আক্ষরিকভাবেই একটা গোটা নাম হিশেবে পড়তে। কেন, সেটা আমরা জানতে পারব ব্যাশ শেল নিয়ে বিশদ আলোচনায়, যদি সেটা আদৌ আমাদের পাঠমালায় আসে অতটা দূর অব্দি। এবার যে কোনো কমান্ডেই মধ্যে এই '--' ডান্ডাটা দেওয়া থাকলে আর কোনো ঝামেলা হবেনা। যেকোনো কমান্ডই কাজ করবে। তবে, সবচেয়ে সহজ হল '`mv -- -essay essay`' করে ফাইলটার নামের হাইফেনটা উড়িয়ে দেওয়া। এখানে বলে রাখি, শুধু হাইফেন না, এই রকম ঝামেলা হতে পারে '$', '`', '?' বা '*' চিহ্ন নিয়েও। এই চিহ্নগুলোর শেলে আলাদা অর্থ আছে। তাই, এই চিহ্নগুলো ফাইলনামে ব্যবহার করলে শেল কুপথে চালিত হওয়ার সম্ভাবনা থাকে। নইলে বারণ নেই এর কোনোটা ব্যবহারেই। যে কোনো ল্যাটিন মানে ইংরিজি অক্ষর, তার বড় বা ছোট হাতে, 'A-Z' বা 'a-z', যে কোনো অঙ্ক, '0-9', পিরিয়ড মানে ডট বা বিন্দু বা '.', আন্ডারস্কোর মানে '_' — এর কোনোটাতেই কোনো সমস্যা নেই।

৩.১।। এক স্তরের সরলতম ডাইরেক্টরি সিস্টেম

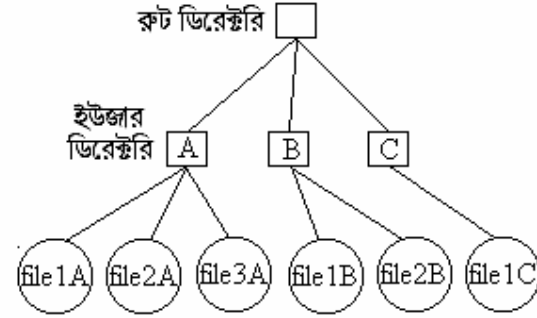
এক একটা সিস্টেমে মোট থাকে হাজার হাজার লাখ লাখ ফাইল। আমার দুটো হার্ড ডিস্কে, মোট সাতটা পার্টিশন, তার মধ্যে একটা সোয়াপ ফাইল, মোট আশি জিবিতে, ফাইলের সংখ্যা ৫৫৬৯১৩। একটা ব্যক্তিগত সিস্টেমেই এই সাড়ে পাঁচ লাখেরও বেশি ফাইল, এবার একটা অফিস বা ইনস্টিটিউট-এর মোট ফাইলের সংখ্যা ভাবুন, এই অগণিত ফাইল থাকার জন্যে, অবশ্যই একটা ডাইরেক্টরি কাঠামো দরকার পড়ে। চার নম্বর দিনের সেকশন ১৩-য় আমাদের আলোচনা থেকে, ১৯৬০-এর দশকের পৃথিবীর সবচেয়ে শক্তিশালী মেইনফ্রেম কম্পিউটার সিডিসি ৬৬০০-র কথা মনে করুন, যাকে 'সুপারকম্পিউটার' বলে ডাকা হয়েছিল, যেখানে যাবতীয় সমস্ত ব্যবহারকারীর জন্যে ডিরেক্টরি ছিল সাকুল্যে একটা — প্রতিটি ইউজারের প্রতিটি ফাইলই থাকত সেই একমেবাদ্বিতীয়ম ডিরেক্টরিতে। ভাবুন তো সেইভাবে যদি রাখতে হত একটা ব্যক্তিগত সিস্টেমেরই ছয় লাখ ফাইল, তাহলে লোকে কাজ করবে কী, ফাইলের আলাদা আলাদা নাম আবিষ্কার করতে গিয়েই তো কাঁধ ঝুলে যাবে। আসলে সিডিসি ৬৬০০-র পরের চল্লিশটা বছরে পৃথিবী, মানুষ, তার কাজ, কাজের হাতিয়ার — সবই বদলে গেছে বিরাট রকমে। তাই এখন ডাইরেক্টরি কাঠামোও দরকার পড়ে অনেক বিশদ জটিল এবং স্তরবিন্যস্ত।

সিডিসি ৬৬০০ গোছের মেশিনে, এবং এর পরেও কিছুদিন অব্দি, ব্যবহৃত হত, এই সরলতম ডিরেক্টরি কাঠামো, যাতে একটাই ডিরেক্টরির উদরে সুখে বসবাস করত আপামর সমস্ত ব্যবহারকারীর যাবতীয় ফাইল। পরবর্তী ডিরেক্টরি কাঠামোর সঙ্গে তুলনা করে আলোচনা করতে গিয়ে এটাকেও রুট ডিরেক্টরি বলে ডাকা হয়, কিন্তু এক ভাবে দেখলে সেটা অর্থহীন। কারণ ডিরেক্টরিই তো সাকুল্যে এক পিস। এখানে এই ডিরেক্টরিটায় আছে ছটা ফাইল।

বাঁদিক থেকে প্রথম তিনটের মালিক ব্যবহারকারী A, ফাইলগুলো হল file1A, file2A, এবং file3A। পরের দুটো ফাইলের মালিক B, ফাইলদুটো file1B এবং file2B। ছ নম্বর ফাইল file1C, তার মালিক C। এখানে আমরা ফাইলের নাম না লিখে শুধু মালিকের নাম লিখলেই পারতাম, কারণ, ফাইলের নামটার এখানে কোনো গুরুত্বই নেই তেমন। একজন ব্যবহারকারীর কটা ফাইল আছে, এবং কত নম্বর ফাইলটা আমরা খুঁজছি, এটাই জরুরি। এবং এখানে সেই খুঁজে পাওয়াটা হবে অসম্ভব দ্রুত, কারণ খোঁজার জায়গাটা তো মাত্র এক।

এই ধরনের ডিরেক্টরি কাঠামোয় বিপজ্জনক জায়গাটা এই যে, কোনো একটা নামে আমি আমার একটা ফাইল লিখেছি, এবার আপনি যদি ওই একই নামে কোনো ফাইল সেভ করেন, সেটা আমার ফাইলটাকে ওভাররাইট করে, সেটার জায়গায় নিজেকে সিস্টেমে ঢুকিয়ে দেবে। এই ধরনের ডিরেক্টরি কাঠামো আমাদের পিসি সিস্টেমে আর ব্যবহার হয়না, হওয়া সম্ভবও না, কিন্তু

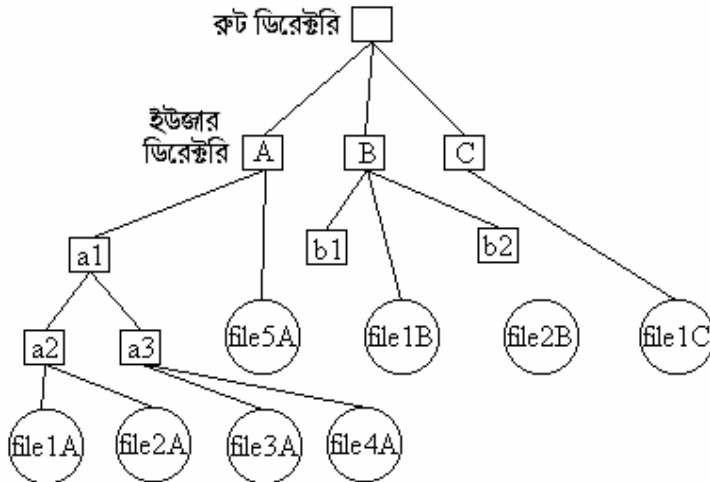
কোনো কোনো এমবেডেড সিস্টেমে, যেখানে প্রতিদিন গ্নু-লিনাক্সের ব্যবহার আগের চেয়ে বেড়ে যাচ্ছে, এটা ব্যবহার হওয়া সম্ভব। ধরুন একটা যন্ত্রের নিরাপত্তা ব্যবস্থা — কে কে সেটা ব্যবহার করতে পারবে, এখানে কয়েকজন ব্যবহারকারীর প্রত্যেকের একটা করে প্রোফাইল ভরে রাখতে হচ্ছে, পরিচয়ব্যবস্থা, হয়তো তাতে ছবি থাকছে, আঙুলের ছাপ থাকছে, সই থাকছে — ইত্যাদি। মূল ব্যাপারটা হল, প্রতিজন পিছু একটা করে ফাইল এবং সেই ফাইলের জায়গায় অন্য কারুর ফাইল সেভ করে দেওয়ার কোনো সম্ভাবনা নেই। যেমন আজ, ১২-ই নভেম্বর, খবরের কাগজে দেখছিলাম, বিভিন্ন জায়গায় এটিএম, বা অটোমেটিক টেলর মেশিন মূল কথাটা, কিন্তু কিরকম করে জানি এনিটাইম-মানি কথাটাই চালু হয়ে গেছে, সেই এটিএম ব্যবস্থা ভেঙে পড়ছে ভাইরাস আক্রমণে, এবং এই ব্যাপারে কথা বলতে গিয়ে একাধিক বিশেষজ্ঞ বলেছেন, এটা উইনডোজে কাজ হচ্ছে বলেই চিন্তা।

৩.২।। দুই স্তরের ডিরেক্টরি কাঠামো

এর পরের স্তরে, বিভিন্ন ব্যবহারকারীর মধ্যে গোলযোগ এড়াতে, এল দুই স্তরের ডিরেক্টরি কাঠামো। এই ছবিটাতেও দেখুন, আমরা চৌকো দিয়ে ডিরেক্টরি আর গোল দিয়ে ফাইল বুঝিয়েছি। এই দুই-স্তরের ডিরেক্টরি কাঠামোয় প্রতিটি ইউজার বা ব্যবহারকারীকে দেওয়া হল তার নিজের নিজের ডিরেক্টরি, যাকে আমরা A, B আর C দিয়ে দেখিয়েছি, তিন জন ব্যবহারকারীর নামে। এখন দেখুন প্রথম জনের তিনটে ফাইল, দ্বিতীয় জনের দুটো ফাইল আর তৃতীয় জনের একটা ফাইল আছে এদের নিজের নিজের ডিরেক্টরিতে। তাই ফাইলনাম এক না আলাদা তাতে আর কিছু এসে যায়না। একাধিক ইউজারের কম্পিউটারে, বা, খুব সরল নেটওয়ার্ক সিস্টেমেও এই কাঠামো ব্যবহার করা যায়।

এই ধরনের ডিরেক্টরি কাঠামোয় সিস্টেমের একটা ভূমিকা চলে এল এই অর্থে যে, যখনই কোনো একজন ব্যবহারকারী একটা ফাইল খোলার চেষ্টা করবে, সিস্টেমের জানা থাকবে, এই ব্যবহারকারী কে, এবং সেই অনুযায়ী তার মালিকানার ডিরেক্টরিতে গিয়ে ফাইলটা খুঁজবে। তার মানে, খেয়াল করুন, এইবার লগ-ইন ধারণাটা আসছে। এর সঙ্গে মেলান দিন চার-এ আমাদের মাল্টিপ্রোগ্রামিং থেকে মাল্টিইউজারে বিবর্তনের আলোচনাটা। লগ-ইন থাকছে মানেই এখানে প্রতিটি ব্যবহারকারীর একটা করে আইডেন্টিটি বা পরিচিতি এবং সেই পরিচিতি প্রমাণের দায় থাকছে, যা এক স্তরের ডিরেক্টরি কাঠামোয় ছিলনা।

এই স্তরের ডিরেক্টরি কাঠামোয় অন্য আর একটা ধরনের প্রয়োজন তৈরি হতে শুরু করল। ধরুন, প্রতিটি ইউজার শুধু তার নিজের ডিরেক্টরির ফাইলই খুলতে পারে, অন্য কোথাও কোনো ফাইলে তার অধিকার নেই। এই অবস্থায় যে সাধারণ কাজগুলো প্রতিটি ব্যবহারকারীকেই করতে হয়, তার প্রোগ্রামগুলোকেও প্রতিটি ব্যবহারকারীর ডিরেক্টরিতেই থাকতে হবে। আমরা যে প্রোগ্রামগুলোকে দেখেছি, /bin ডিরেক্টরিতে থাকতে, ls rm cp ইত্যাদি, সেগুলোর একটা করে কপি থাকতে হবে প্রতিটি ব্যবহারকারীর ডিরেক্টরিতেই। এটা একটা সিস্টেমের ইনএফিশিয়েন্সি, ক্যাবলামো। কাজের পরিমাণ, প্রোগ্রামের পরিমাণ যত বাড়ে এই ক্যাবলামোটা ততই আরো ব্যয়বহুল হয়ে পড়ে। অর্থাৎ, এমন একটা সিস্টেম ডিরেক্টরির দরকার পড়ছে যার মধ্যে থাকবে একজিকিউটেবল বাইনারি প্রোগ্রামগুলো — যা যে কোনো ইউজার ব্যবহার করতে পারবে।

৩.৩।। ক্রমানুসারী ডিরেক্টরি কাঠামো

এবার এল হায়েরার্কিকাল ডিরেক্টরি স্ট্রাকচার বা ক্রমানুসারী ডিরেক্টরি কাঠামো। দুই স্তরের কাঠামোয় নাম নিয়ে ঝামেলাটা চলে যায়, কিন্তু বহু ইউজারের বহু ফাইলের একটা ব্যবস্থায় খুব একটা সুবিধে হয়না। এমনকি একটা ব্যক্তিগত পিসি সিস্টেমেও খুব একটা কাজের নয়, আরো কাজের পরিমাণটা তো প্রতিদিনই বাড়ছিল। ইউজার তার কাজের যুক্তি মাফিক তার নিজের ফাইলগুলোকে আলাদা আলাদা ডিরেক্টরিতে রাখতে চাইবেই, যেমন

আপনার ওই প্রবন্ধ আর বইয়ের চক্করটা — এই রকম প্রয়োজন ক্রমে তো বাড়তেই থাকে, লেখার ফাইল, গানের ফাইল, কোডের ফাইল, কম্পাইল করার পর বাইনারি একজিকিউটেবল ফাইল, ইমেল ফাইল, ওয়েবপেজ, এইরকম। এই হায়েরার্কিকাল ডিরেক্টরি কাঠামোয় সেটা করার সুযোগ এল তার। এটায় দেখুন, সিস্টেমের বরাদ্দ করা তিনটে ইউজার ডিরেক্টরিকে আমরা দেখিয়েছি তিনজন ব্যবহারকারী — A, B আর C — তাদের নিজেদের নামেই। আর তাদের নিজেদের তৈরি করা ডিরেক্টরিগুলোকে দেখিয়েছি ছোট হাতে, a1 a2 a3 b1 b2 — এইরকম। এখানে দেখুন, A তার নিজের ইউজার ডিরেক্টরির মধ্যে a1 ডিরেক্টরি তৈরি করেছে। a1-এর মধ্যে আবার সাবডিরেক্টরি a2 আর a3। যার প্রত্যেকটায় দুটো করে মোট চারটে ফাইল। A-র পাঁচ নম্বর ফাইলটা আছে সরাসরি ইউজার ডিরেক্টরিতেই। ঠিক এই ভাবে যে কোনো ব্যবহারকারী তার নিজের ইউজার ডিরেক্টরিতে এবং তার মধ্যের যে কোনো সাবডিরেক্টরিতে যত খুশি ডিরেক্টরি এবং ফাইল বানাতে বা রাখতে পারে। একটা আধুনিক কম্পিউটার ব্যবস্থায় এই হায়েরার্কিকাল ডিরেক্টরি কাঠামোই ব্যবহৃত হয়।

৪।।পথনির্দেশ

এবার ডাইরেক্টরি কাঠামোয় সাজানো একটা ফাইল ব্যবস্থায় আমরা একটা জায়গা থেকে অন্য জায়গায় যাবার পথনির্দেশ করব কী করে? এর দুটো আলাদা উপায় আছে। একটা হল চূড়ান্ত পথনির্দেশ বা অ্যাবসলিউট পাথ। আর অন্যটা আপেক্ষিক পথনির্দেশ বা রিলেটিভ পাথ। ধরুন, এতক্ষণ ধরে, কোনো ডিরেক্টরি বলতে হলেই আমরা যে শুধু তার নাম না বলে সঙ্গে একটা স্ল্যাশ জুড়ে দিচ্ছিলাম, এই আগের আগের প্যারাগ্রাফেই দেখুন বাইনারি ফাইলগুলো কোথায় থাকে বলতে গিয়ে, 'bin' না-বলে, বললাম '/bin'। কেন? এটা বোঝাতে যে ওই 'bin' ডিরেক্টরিটা আছে '/'-এর মধ্যে, যা হল একটা গ্নু-লিনাক্স কাঠামোয় রুট ডিরেক্টরি, মানে আর সমস্ত ডিরেক্টরিরাই যার সাবডিরেক্টরি, বা সাবডিরেক্টরির সাবডিরেক্টরি। চূড়ান্ত পথ বা অ্যাবসলিউট পাথটা শুরুই হয় এই রুট ডিরেক্টরি স্ল্যাশ '/' থেকে। আবার এই স্ল্যাশ-ই ব্যবহার হয় একটা মা-ডিরেক্টরির সঙ্গে তার ছানা-ডিরেক্টরির সম্পর্ক বোঝাতে। যেমন ধরুন, আমরা বললাম '/usr/local/bin'। এর মানে, মূল মা ডিরেক্টরি বা রুট ডিরেক্টরি '/', তার ছানা হল 'usr', তার ছানা 'local', তার ছানা 'bin', সেই ডিরেক্টরির কথা বলছি আমরা — যার চূড়ান্ত ঠিকানা '/usr/local/bin'। এবার, যেই চূড়ান্ত ঠিকানাটা দিয়ে দিলাম, '/usr/local/bin' আর '/bin' আলাদা ডিরেক্টরি বলে চেনা গেল, দুজনের একই নাম 'bin' হওয়া সত্ত্বেও।

চূড়ান্ত বা অ্যাবসলিউট পথ তাই সবসময়ই শুরু হয় চূড়ান্ত বিন্দু বা মেরু থেকে, মানে রুট ডিরেক্টরি, গ্নু-লিনাক্সে যাকে আমরা লিখি স্ল্যাশ (/) দিয়ে, আর ডিরেক্টরি থেকে সাবডিরেক্টরি ভিন্নক বা সেপারেটরও ওই স্ল্যাশ। যাবতীয় ইউনিক্স মেক রিমেক মিক্স রিমিক্সেই এটা প্রথা। উইন্ডোজের সেপারেটরটা হল ব্যাকস্ল্যাশ (\), মাল্টিক্স-এ ছিল 'গ্রেটার দ্যান' বা 'বৃহত্তর' চিহ্ন (>)। এবার ধরুন, আমরা বলছি, '' ডিরেক্টরির মধ্যে আছে '' ডিরেক্টরি, তার মধ্যে আছে '' ডিরেক্টরি, তার মধ্যে আছে '' ডিরেক্টরি, যার কথা আলোচনা করছি আমরা। এই পথনির্দেশটা গ্নু-লিনাক্স তথা যেকোনো ইউনিক্স, উইনডোজ আর মাল্টিক্স, এই তিন রকম প্রথায় লেখা হত তিন ভাবে —

গ্নু-লিনাক্স — /usr/local/share/doc/

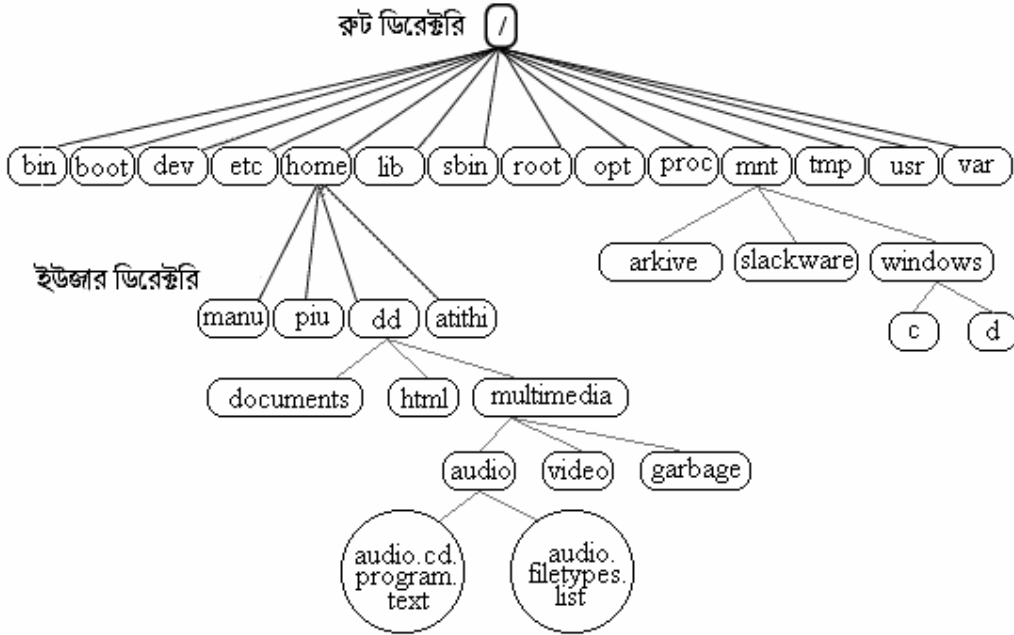
উইনডোজ — \usr\local\share\doc\

মাল্টিক্স — >usr>local>share>doc>

এখানে উইনডোজ পাথটা কিন্তু কোনো চূড়ান্ত পথ বা অ্যাবসলিউট পাথ নয়, কারণ, উইনডোজ '\' থেকে শুরু হয়না। উইনডোজ থেকে গ্নু-লিনাক্সে রুট ডিরেক্টরির ধারণাটা একদম আলাদা, সেটা আমরা পরে আরো ভালো করে বুঝতে পারব, মাউন্ট বোঝার সময়, এমনকি যদি 'C:' তার রুট ডিরেক্টরি বলে ধরে নিই, তাহলেও সেটা এখানে উল্লেখিত নেই। কিন্তু, গ্নু-লিনাক্স-এর বেলায় '/' দিয়ে শুরু মানেই সেটা অ্যাবসলিউট বা চূড়ান্ত পথ। সরাসরি রুট ডিরেক্টরি থেকে পুরো পথটা দেখাচ্ছে — যে অব্দি আমরা যেতে চাই।

অন্য রকমের পথনির্দেশটা হল আপেক্ষিক বা রিলেটিভ। আপেক্ষিক মানেই তার মধ্যে নিহিত আছে 'ওয়ার্কিং ডিরেক্টরি' বা 'কাজের ডিরেক্টরি' বলে একটা ধারণা, আপেক্ষিক বলতে সেই ওয়ার্কিং ডিরেক্টরির সাপেক্ষে আপেক্ষিক। এই শব্দবন্ধ 'ওয়ার্কিং ডিরেক্টরি' একবারই এসেছে আমাদের এই লেখায়, কোথায় মনে করতে পারছেন? একটা কমান্ডে, যে কমান্ডটা আপনি ব্যবহার করেছেন? কমান্ডটা ছিল, 'প্রিন্ট ওয়ার্কিং ডিরেক্টরি', কাজের ডিরেক্টরি দেখাও, pwd, ব্যবহার করে আপনি দেখে নিয়েছিলেন রিডাইরেক্ট করা audio.cd.program.text ফাইলটা কোথায় সেভ করা হল। সেই '/home/dd/multimedia/audio' ডিরেক্টরি।

এবার যেই ওয়ার্কিং ডিরেক্টরি বলে একটা কিছু আছে, অমনি আমাদের পথ দেখানোর উপায়টাও তার সাপেক্ষে বদলে যাবে। প্রথমে ফাইলদুটোকে দেখে নিন, গোল চিহ্নে, audio.cd.program.text আর audio.filetypes.list, যারা আপনাকে কিছুতেই ব্যাশ শেলে টাইপ করার ট্যাবারাম পেতে দিচ্ছিল না দুটোরই শুরুতে ওই 'audio.' থাকায়। ফাইলদুটোর বজ্জাতি এখানেই শেষ নয়, ছবিতে ডিরেক্টরির ছকগুলো বানিয়েছিলাম আমীর সাইজের, তার মধ্যে এই বচ্চন সাইজের ফাইল, কী বিকট দেখাচ্ছে, কিন্তু, সরি, কিছু করার নেই, আজ তেরোই ডিসেম্বর, শনিবার, পরশু জয়েন করছি, এখন আর সময় নেই নতুন করে এতগুলো

পাতা ধরে ফাইলনাম বদলে আসার, এবং ভাঙার পক্ষে আমার আরো একটা ঠ্যাং এই মুহূর্তেই তৈরি আছে ঠিকই, হাত নয় কারণ লেখা যাবেনা, কিন্তু আর ছুটি নেই। ইন ফ্যাক্ট যা শুনছি, এতেই বোধহয় উইদাউট পে — তখন তো আপনাদের পাবনা। এবার ওই বারো হাত কাঁকুড়ের তেরো হাত বীচি ব্রান্ডের ফাইলদুটোকে দেখুন। এর অ্যাবসলিউট বা চূড়ান্ত ঠিকানাটা ভাবুন, শুরু হয়েছে একদম উপরের রুট ডিরেক্টরি স্ল্যাশ '/' থেকে, তারপর 'home', তারপর 'dd' মানে একজন ব্যবহারকারী, এইটা হল ইউজার ডিরেক্টরি, আগের হায়েরার্কির ছকের সঙ্গে মেলান, তার মধ্যে 'multimedia', তার মধ্যে 'audio' — এইখানে রয়েছে 'audio.cd.program.text'।

তাহলে চূড়ান্ত পথনির্দেশটা মেলানো গেল '/home/dd/multimedia/audio/audio.cd.program.text'। এবার এটাকে আপেক্ষিক ভাবে ভাবুন। ধরুন, আগের দিন ফাইলটাকে সেভ করে বেরিয়ে গেছি। এবার পরের দিন লগ-ইন করলাম। মনে আছে সেই পুণ্যের কাজের কথা — পথহারা ব্যবহারকারীকে তার ঘর দেখায় লগ-ইন, পাঁচ নম্বর দিনে আমরা লগ-ইনের কাজে আলোচনা করেছি। এবার, আমি লগ-ইন কোথায় করেছি? আমার ইউজার ডিরেক্টরিতে, মানে '/home/dd'। ঢুকেই আমি '`pwd`' কমান্ডটা লিখে যদি এন্টার মারি, কমান্ড প্রম্পটে ফুটে উঠবে '`/home/dd`'। এবার এই '' ডিরেক্টরিতে আমার ঠিক সামনে কী আছে? তিনটে ডিরেক্টরি — documents, html, এবং multimedia। আমি যেতে চাই 'multimedia' ডিরেক্টরির মধ্যে 'audio' ডিরেক্টরিতে। এইজন্যে এবার আমি যে কমান্ডটা ব্যবহার করব, তার নাম 'chdir'।

আমি কমান্ড দেব '`chdir multimedia`'। এবার আবার '`pwd`' করলে ও দেখাবে '`/home/dd/multimedia`'। এবার ভাবুন তো, যার ঠিকানাটা '`/home/dd/multimedia`', সেখানে আমি পৌঁছে গেলাম শুধু '`chdir multimedia`' করে। এখানে আমি যদি '`chdir /home/dd/multimedia`' করতাম, তাহলেও একই কাজ হত, কিন্তু তার মানে তখন আমি চূড়ান্ত বা অ্যাবসলিউট পথ ব্যবহার করছি। আমার ওয়ার্কিং ডিরেক্টরির সাপেক্ষে রিলেটিভ পথ নয়। এখানে সরাসরি শুধু '`chdir multimedia`' করেই কাজ হল, কারণ ওই ডিরেক্টরিটা ঠিক আমার ওয়ার্কিং ডিরেক্টরিতেই বিরাজ করছে। এবার এই /home/dd/multimedia ডিরেক্টরিতে দাঁড়িয়ে আমরা সেই 'mv' কমান্ডটার অন্য কাজটাও করতে পারব, যা আগেরবার পারিনি, কারণ তখন আমরা এই পথনির্দেশ ব্যাপারটা জানতাম না। আগেরবার আমরা শুধু একটা ফাইলের নাম বদলেছিলাম, এবার একটা ফাইলকে নড়াব, তখন এটা পারিনি, কারণ, এটা করা মানেই ডিরেক্টরি কাঠামোর ভিতর নতুন ঠিকানাটা আমায় দিতে হবে। আমি এখন রয়েছি /home/dd/multimedia ডিরেক্টরিতে, এখানে আমার ওয়ার্কিং ডিরেক্টরিতে রয়েছে, ছবিতে দেখুন, audio, video, আর garbage নামে তিনটে ডিরেক্টরি। এই audio ডিরেক্টরির মধ্যে আছে সেই ঝামেলা পাকানো ফাইল audio.filetypes.list। এবার তাকে আমরা নড়িয়ে garbage ডিরেক্টরিতে নিয়ে যাব। আমরা কমান্ড দেব '`mv audio/audio.filetypes.list garbage/audio.filetypes.list`'। আমরা ব্যাবহার করলাম আপেক্ষিক বা রিলেটিভ পথ, কারণ এই ডিরেক্টরিতে আমার সামনেই রয়েছে audio আর garbage ডিরেক্টরিদুটো, তার একটার থেকে একটা ফাইল নিয়ে আমি অন্য ডিরেক্টরিতে রাখতে বললাম। এতক্ষণ আমরা '`ls`' কমান্ড দিয়েছি যে ডিরেক্টরির

ফাইল দেখছি সেই ডিরেক্টরিতে দাঁড়িয়ে, কারণ আমরা পথনির্দেশ জানতাম না, এবার আমরা এখানে দাঁড়িয়েই কমান্ড দেব — '`ls garbage`', এবং দেখতে পাব ফাইলটা ওই ডিরেক্টরিতে চলে গেছে। এই রিলেটিভ পথেই কাজ চলে গেল, কারণ, এই garbage ডিরেক্টরিটা আমার ওয়ার্কিং ডিরেক্টরিতেই আছে। যদি আমি /home/dd ডিরেক্টরিতে দাঁড়িয়ে থাকতাম, কমান্ড দিতাম '`ls multimedia/garbage`'। এখানেও হত রিলেটিভ পথ, একটু বড় রিলেটিভ পথ, কারন যে ডিরেক্টরিটা আমি দেখতে চাইছি, সেটা ওয়ার্কিং ডিরেক্টরিরই একটা সাবডিরেক্টরিতে আছে যার নাম multimedia। ধরুন, আমরা যদি এই পথের সম্পূর্ণ বাইরে, রুট ডিরেক্টরির নিচেই সবচেয়ে বাঁদিকের সাবডিরেক্টরিটায় থাকতাম, মানে '/bin' ডিরেক্টরিতে, তাহলে আমাদের ওই একই ফাইল দেখার জন্যে কমান্ড দিতে হত '`ls /home/dd/multimedia/garbage`'। এটা হত চূড়ান্ত বা অ্যাবসলিউট পথ।

৫।। ঐক্যবদ্ধ ফাইল সিস্টেম

আমাদের পথনির্দেশ জানা হল, কী করে একটা হায়েরার্কিকাল বা ক্রমানুসারী ডিরেক্টরি কাঠামোয় একটা ফাইলের বা ডিরেক্টরির ঠিকানা দিতে হয়। এবার আমরা যাব, এই ডিরেক্টরি কাঠামোর গ্নু-লিনাক্স আকার ঠিক কী হয় তার আলোচনায়। তার আগে, এই শেষ ছবিটায় ঠিক রুট ডিরেক্টরির উপরে প্রথম স্তরের সাবডিরেক্টরিগুলো খেয়াল করুন, /bin, /boot, /dev, /etc, /home, /lib, /sbin, /root, /opt, /proc, /mnt, /tmp, /usr, /var। এগুলো কিন্তু আনতাবড়ি নয়, আমার সুজে ৮.২ সিস্টেমের সত্যিকারের ডিরেক্টরি কাঠামোটাই দিয়েছি। এই ছবিটা একটু মন দিয়ে দেখে রাখুন, এরপর বারবার এটার কথা আসবে। '/' এবং তার পরেই তার থেকে প্রবাহিত এই ডিরেক্টরিগুলো — এই গোটাটা মিলিয়ে তৈরি হয় গ্নু-লিনাক্সের ইউনিফায়েড বা ঐক্যবদ্ধ ফাইল সিস্টেম। খেয়াল করুন, এটা একটা সিঙ্গল হায়েরার্কিকাল স্ট্রাকচার বা এক-উৎসের ক্রমানুসারী ডিরেক্টরি কাঠামো। একটা উৎস, রুট ডিরেক্টরি '/' — তার থেকে প্রবাহিত হয়েছে আর সমস্ত ডিরেক্টরি। আপনি যে ফাইল যে ডিরেক্টরিই ভাবুন না কেন তা এই মূল উৎস থেকে প্রবাহিত কোনো না কোনো ধারায়, বা কোনো না কোনো উপধারায় আছে।

উইন্ডোজ থেকে গ্নু-লিনাক্সে অভিবাসীরা প্রথম প্রথম তাদের হারিয়ে যাওয়া ড্রাইভ নিয়ে হাপিত্যেশ করেন, কোথায় গেলি C:\ মানে সি-ড্রাইভ, বা D:\ বা A:\ ইত্যাদি। এখানে তেমন কোনো ড্রাইভের গল্পই নেই। ওইরকম আলাদা আলাদা ড্রাইভ আছে বলে উইন্ডোজ হল একটা মাল্টিপল হায়েরার্কিকাল ডিরেক্টরি স্ট্রাকচার, বা বহু-উৎসের ক্রমানুসারী ডিরেক্টরি কাঠামো। সি-ড্রাইভ, ডি-ড্রাইভ, এ-ড্রাইভ, এই সবগুলোরই নিজের নিজের এক একটা রুট বা মূল ডিরেক্টরি আছে। যেখানটা নিছক C:\, এবার যেই আপনি উইন্ডোজ ডিরেক্টরিতে ঢুকলেন সেটা হয়ে গেল, C:\Windows\ — যদি আপনি এমএসডস-প্রম্পটে থাকেন তাহলে তো বটেই, উইন্ডোজ এক্সপ্লোরারে থাকলেও দেখবেন লোকেশন বারে, মানে একদম উপরের দিকে, মেনুবারের নিচেই সাদা জায়গাটায়, এই ঠিকানাটা ফুটে উঠছে, যেই আপনি বাঁদিকের প্যানেলে উইন্ডোজ ডিরেক্টরির উপর ক্লিক করলেন, এবং ডানদিকে ফুটে উঠল সেই ডিরেক্টরির মধ্যের সাবডিরেক্টরি এবং ফাইলগুলো। এরকম রুট ডিরেক্টরি উইনডোজে থাকে একটা করে, কিন্তু গ্নু-লিনাক্সে তথা যে কোনো ইউনিক্সে রুট ডিরেক্টরি হল এক এবং অদ্বিতীয় '/'। এইজন্যে গ্নু-লিনাক্স ফাইলসিস্টেমকে একটা ঐক্যবদ্ধ ফাইল সিস্টেম বলা হয়।

একটু বাদেই আমরা দেখব একটা মেশিনের কোন হার্ড ডিস্কের কোন কোন পার্টিশনকে এই ঐক্যবদ্ধ ফাইল সিস্টেমের ভিতরে নিয়ে আসা হবে সেটা নিজের ইচ্ছের উপর নির্ভর করে। একে টেকনিকাল পরিভাষায় আমরা বলি মাউন্ট করা। যে যে পার্টিশনে মাউন্ট করা হয় সেই সেই পার্টিশনের নিজের ফাইল সিস্টেম হায়েরার্কি এবার গোটা সিস্টেমের ফাইল সিস্টেম হায়েরার্কির একটা উপধারা হয়ে পড়ে। ধরুন, ওই সি-ড্রাইভ বা C:\ যে পার্টিশনে আছে তাকে মাউন্ট করা হয়েছে সেটা হল /mnt/windows/c/, এবার ঠিক সেই সেই ফাইল এবং ডিরেক্টরি সেই সেই সজ্জায় আমি পাব এই /mnt/windows/c/ ডিরেক্টরিতে যাদের যেভাবে আমি পেতাম C:\ ড্রাইভে। অর্থাৎ ওই সি-ড্রাইভের নিজের ফাইলসিস্টেম হায়েরার্কিটা এবার রুট ডিরেক্টরি '/'-এর ভিতর একটা উপধারা হয়ে গেল, মানে একটা সাবসেট হয়ে গেল ঐক্যবদ্ধ ফাইল সিস্টেমের সিঙ্গল হায়েরার্কির। এখন থেকে সি-ড্রাইভ যে ভৌত পার্টিশনে আছে সেই পার্টিশনে রাখা সমস্ত কিছুকে আমরা পাব /mnt/windows/c/, সেইজন্যে এই ডিরেক্টরিটা হল ওই পার্টিশনের মাউন্ট পয়েন্ট। এই ভাবে যে কোনো পার্টিশনেই মাউন্ট করা যায়, মানে তাকে নিজের ঐক্যবদ্ধ ফাইল সিস্টেমের মধ্যে ঢুকিয়ে আনা যায়। এমনকি সেই পার্টিশন যদি নেটওয়ার্কে সংযুক্ত অন্য কোনো মেশিনেও থাকে।

এমনকি একটা পার্টিশনকে দুটো সিস্টেম শেয়ারও করতে পারে। মানে দুজনেই ব্যবহার করতে পারে। কিছু কিছু ডিরেক্টরির বেলায় পারেনা, যেমন /boot, /etc, বা /var — এই ডিরেক্টরিগুলো যেখানে মাউন্ট করা সেই পার্টিশন দুটো সিস্টেম একই সাথে ব্যবহার করতে পারেনা। /etc ডিরেক্টরিতে থাকে সিস্টেমের সমস্ত কনফিগারেশন ফাইল, যেমন পাঁচ নম্বর দিনেই দেখেছেন আপনার সিস্টেমের প্রতিটি পাসওয়ার্ড থাকে এই ডিরেক্টরিতেই, সেটা অন্য সিস্টেমের সঙ্গে শেয়ার করবেন কী

করে? /boot ডিরেক্টরিতে থাকে আপনার সিস্টেমের কারনেল এবং তার খুঁটিনাটি, সেটাও সম্ভব না। একটু পরেই, লিনাক্স ফাইল সিস্টেম হায়েরার্কির সর্বজনগ্রাহ্য নিরিখ বা স্ট্যান্ডার্ডের আলোচনায় আমরা এগুলো ভালো করে জানতে পারব।

এই ঐক্যবদ্ধ ফাইল সিস্টেমের আর একটা গুরুত্বপূর্ণ জায়গা হল গ্নু-লিনাক্সে অত্যুন্নত মেমরি ব্যবহার ব্যবস্থা, যার অন্য নাম ক্যাশে। গোটা মেমরি ব্যবহারের প্রক্রিয়াটাকেই যা অনেকটা দ্রুততর করে তোলে। শূন্য এক আর দুই নম্বর দিনে বারবার আমাদের আলোচনায় এসেছে, কম্পিউটারে মূলত দুই রকমের মেমরির কথা, র‍্যাম আর রম। এই র‍্যামকে আবার দুইভাবে দেখেছি আমরা, ভৌত এবং ভৌতিক, ফিজিকাল এবং ভারচুয়াল। এই ভারচুয়াল মেমরি, সোয়াপফাইল, ক্যাশে — গোটা ব্যবস্থাটাই চলে একটা বিপুল পরিমাণ তথ্য অস্থায়ী ভাবে ভৌত ডিস্কে তুলে রাখার ভিত্তিতে। এই ব্যবস্থাটা এত ধারালো বলে গ্নু-লিনাক্সে মেমরির ব্যবহার চমৎকার। উইনডোজ ব্যবস্থায় র‍্যাম খুব বেশি বাড়ালে যেমন আদতে সিস্টেমের গতি কমেও যেতে পারে, যেটা ঘটে সেটা বাংলায় বললে এই যে সিপিইউ-র স্ক্রু ঢিলে হয়ে যায় ওই স্তূপ স্তূপ মেমরির পাতার মধ্যে পড়ে, কী খুঁজছে সেটাই হারিয়ে ফেলে। ফাইনালের আগের দিন রাত্তিরে ধরুন আপনাকে দেওয়া হল, কালকের কোশ্চেন আসতে পারে মাত্র এই একুশটা বই থেকে, পড়ে ফেলো বাবা। পরের দিন আপনি হলে গিয়ে, কুল বাংলার কোশ্চেনের উর্দুর উত্তর লিখে চলে এলেন।

গ্নু-লিনাক্স আসলে ঘুঘু জিনিষ, সে গোটাটা পড়েই না, টুকলি বানাতে থাকে। এবার, ঐক্যবদ্ধ যদি না হয় গোটা সিস্টেমটা, তাহলে, আবার সেই সমস্যা, কোথাকার টুকলি কোথায় যাবে, কোন মেমরি বাফার কোথায় লেখা হবে। এই জন্যে একটা গ্নু-লিনাক্স সিস্টেমে সঠিক শাট-ডাউনটা খুব জরুরি। হঠাৎ করে বিদ্যুৎ চলে গেলে বা অফ করে দিলে সমস্যা কিছু হতে দেখিনি আমি কখনো, বা শোনাও যায়না তেমন, গ্নু-লিনাক্সের ফাইল সিস্টেম এত জোরালো বা রোবাস্ট। কিন্তু সিস্টেমের উপর প্রচণ্ড চাপ পড়ে। অন্য অনেক উপায় আছে শাট-ডাউনের, ধীরে ধীরে আপনি সবগুলোই শিখে যাবেন। তবে একটা বেশ সহজ এবং নিরাপদ উপায় আছে, কন্ট্রোল-অল্ট-ডিলিট মারা, তারপর আঙুল উঁচিয়ে বসে থাকা, ঠিক নতুন করে বুট করে যাবার আগেই অফ করে দেওয়া। এটা গুই বা এক্স-উইনডোজে থাকলে হবেনা, সেখানে লগ-আউট করে তারপর কন্ট্রোল-অল্ট-ডিলিট মারতে হবে। এক্স-উইনডোজ থেকে বেরোনোর একটা শর্টকাট হল কন্ট্রোল-অল্ট-ব্যাকস্পেস, তবে সেটা ব্যবহার না-করাই ভালো।

এই ঐক্যবদ্ধ ফাইল সিস্টেমের ছবিটার /home ডিরেক্টরির মধ্যে দেখুন, আবার চারটে ইউজার ডিরেক্টরি, /home/atithi, /home/manu, /home/piu এবং /home/dd। পাঁচ নম্বর দিনে দেখার জন্যে তুলে দেওয়া /etc/passwd ফাইল থেকে মিলিয়ে নিন। আর, একটু মন দিয়ে খেয়াল করুন /root ডিরেক্টরিটা। এটা কিন্তু গোটা হায়েরার্কির চূড়ান্ত মা-ডিরেক্টরি মানে স্ল্যাশ / নয়। এটা হল সুপারইউজার বা রুট নামক ব্যবহারকারীর হোম ডিরেক্টরি। অন্য সব ইউজারের হোম ডিরেক্টরি হয় /home ডিরেক্টরির ভিতর। শুধু এর হোমটাই হয় অন্য জায়গায়। এই যে ডিরেক্টরি কাঠামোটা আমি এখানে দেখিয়েছি, এটা গ্নু-লিনাক্সের একটা বিশেষ ফ্লেভারের, যার নাম সুজে ৮.২, ডিস্ট্রিবিউশনটা সুজে, তার ৮.২ তম ভার্শন। এই ডিরেক্টরি কাঠামোটা, তার খুঁটিনাটি চেহারায়, এমনকি তার উপর উপর অবয়বেও, ফ্লেভার থেকে ফ্লেভারে বেশ বদলায়। আবার একটা একদেশতাও থাকে, গ্নু-লিনাক্স বলে নিজেকে ডাকতে গেলে কিছু ফাইল সিস্টেম হায়েরার্কি স্ট্যান্ডার্ড তাকে মানতে হয়। কিন্তু সেই ন্যূনতম জায়গাটুকুকে বাদ দিয়ে অনেক কনফিগারেশন বা আকারায়ন একটা থেকে অন্যটায় বদলায়। তাই, এই কাঠামোটা মুখস্থ না রেখে এর ধারণাটা মাথায় রাখার চেষ্টা করাই ভালো। এর কোনটার কী মানে, কেন থাকে, এইসবে আসছি — তার আগে এখানে একটা তালিকা দিয়ে রাখি, কত ফ্লেভার আছে আলাদা আলাদা গ্নু-লিনাক্সের, এটা আমি খুঁজে বার করিনি, ২৪ ঘন্টার হাইস্পিড কানেকশন না-থাকলে ওসব করা যায়না, একটা ভারি চমৎকার বই থেকে তুলে দিচ্ছি। বইটার নাম 'ফ্রি ফর অল — হাউ লিনাক্স অ্যান্ড দি ফ্রি সফটওয়ার মুভমেন্ট আন্ডারকাট দি হাইটেক টাইটানস', লেখকের নাম পিটার ওয়েইনার, http://www.wayner.org/books/ffa/ — বইটায় কোনোরকম প্রতিবন্ধ নেই, যে চাইবে তাকেই দেওয়া যাবে, আপনি নিজেও নামিয়ে নিতে পারেন, ১.৩ এমবি।

বইটা শুরু হচ্ছে নিরানব্বই-এর সেই কুখ্যাত গেটস-মামলা দিয়ে — মাইক্রোসফট গোটা সফটওয়ার ব্যবসায় মনোপলি তৈরি করে অন্য সবার ব্যবসা করার আয় করার উপায় বন্ধ করে দিচ্ছে কিনা। এই মামলায়, সে যে মনোপলি নয়, তার যথেষ্ট ভয়াল সব প্রতিদ্বন্দ্বী আছে এটা বোঝাতে গিয়ে গেটস-এর উকিল উল্লেখ করেছিল লিনাক্স এবং এফএসএফ-এর কথা। পিটার ওয়েইনার এই গোটা রসিকতাটার অভ্যন্তরস্থ বিকটতাটাকে চমৎকার দেখিয়েছেন — প্রশ্নটা এখানে টাকা, ক্যাশ, অন্য সবার টাকা আয় বন্ধ করে দেওয়ার জন্যে নোংরামি তুমি করছ কি করছ না, আর তাতে উদাহরণ দিচ্ছ লিনাক্স আর এফএসএফের যারা প্রথম থেকেই বিনা পয়সায় দিয়ে দিচ্ছে গোটাটা, এবং শুধু নিজেদের সফটওয়ারকে নিয়ন্ত্রণ করছে না তাই নয়, সফটওয়ার-প্রকৌশল-কম্পিউটার-চিন্তার গোটা প্রবাহটা যাতে কিছুতেই নিয়ন্ত্রণ করা না-যায় সেই ব্যবস্থা করার চেষ্টা করছে তাদের জিপিএল লাইসেন্স দিয়ে, যাকে অনেকেই মজা করে ডাকে 'কপিলেফট'। বইটা পড়ুন, সত্যিই ভালো লাগবে।

এই বইটা থেকেই তুলে দিই লিনাক্স ডিস্ট্রিবিউশনের তালিকাটা, এবং ইংরিজি অক্ষরেই দিচ্ছি, কারণ এর অনেকগুলোর নামই বাপের জন্মে শুনিনি আমি, আর অনেকগুলোই হল ইউরোপিয় নাম। একদম ল্যাটিন বর্ণমালার অনুক্রমে, Alzza Linux, Apokalypse, Armed Linux, Bad Penguin Linux, Bastille Linux, Best Linux (Finnish/Swedish), Bifrost, Black Cat Linux (Ukrainian/Russian), Caldera OpenLinux, CCLinux, Chinese Linux Extension, Complete Linux, Conectiva Linux (Brazilian), Debian GNU/Linux, Definite Linux, DemoLinux, DLD, DLite, DLX, DragonLinux, easyLinux, Enoch, Eridani Star System, Eonova Linux, e-smith server and gateway, Eurielec Linux (Spanish), eXecutive Linux, floppyfw, Floppix, Green Frog Linux, hal91, Hard Hat Linux, Immunix, Independence, Jurix, Kha0s Linux, KRUD, KSI Linux, Laetos, LEM, Linux Cyrillic Edition, LinuxGT, Linux-Kheops (French), Linux MLD (Japanese), LinuxOne OS, LinuxPPC, LinuxPPP (Mexican), Linux Pro Plus, Linux Router Project, LOAF, LSD, Mandrake, Mastodon, MicroLinux, MkLinux, muLinux, nanoLinux II, NoMad Linux, OpenClassroom, Peanut Linux, Plamo Linux, PLD, Project Ballantain, PROSA, QuadLinux, Red Hat, Rock Linux, RunOnCD, ShareTheNet, Skygate, Slackware, Small Linux, Stampede, Stataboware, Storm Linux, SuSE, Tomsrtbt, Trinux, TurboLinux, uClinux, Vine Linux, WinLinux 2000, Xdenu, XTeamLinux, and Yellow Dog Linux। ওয়েইনারও লিখেছেন, এটা সমাপ্ত কিনা জানিনা, আর আমি তো জানিই-না। তবে যেকটার নাম জানি তার মধ্যে মাংকি লিনাক্স ছাড়া সবকটাই আছে। তবে মাংকিটা নিজেই কোনো ডিস্ট্রিবিউশান না একটু বদলে নেওয়া তা জানিনা, যেমন নপিক্স (Knoppix) হল ডেবিয়ান ডিস্ট্রিবিউশানটাকে একটু বদলে নেওয়া। তবে ওয়েইনারের বইটা ২০০০-এ বেরোনো আর গ্নু-লিনাক্সের মানচিত্রটা এত দ্রুত বদলায়, কারণ গ্রহজোড়া কমিউনিটির অগণ্য মানুষের জ্যান্ত মনগুলো বদলায়, আর মহাভারতের ধর্ম-যখের সেই প্রশ্নোত্তর, মন তো সবচেয়ে দ্রুতগতি, সার্কিটের মধ্যে বিচরমান তথ্যবিদ্যুতের চেয়েও। এবার ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড, এখানে যা আমি জাস্ট ছুঁয়ে যাব, গোটা বিষয়টা খুবই বড় এবং বেশ ইন্টারেস্টিং। এর দুটো চমৎকার ডকুমেন্ট আছে, রাস্টি রাসেল আর ড্যানিয়েল কুইনলানের, 'ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড', অন্যটা বিন এনগুয়েনের, 'লিনাক্স ফাইলসিস্টেম হায়েরার্কি', পরেরটা লিনাক্স ডকুমেন্টেশন প্রোজেক্ট www.tldp.org থেকে পাবেন। আর ট্যানেনবম এবং সুমিতাভ দাস তো আছেই।

৬।। ফাইলসিস্টেমের মধ্যে আলাদা আলাদা ফাইলসিস্টেম

দুয়েকটা ধারণা আগে স্পষ্ট করে নেওয়া যাক। ফাইলকাঠামো বা ফাইলসিস্টেম বলতে আমরা বোঝাচ্ছি, একটা ডিস্কে বা পার্টিশনে রাখা ফাইলগুলোর হিশেব রাখার জন্যে অপারেটিং সিস্টেম যে তথ্য কাঠামো ব্যবহার করে এবং যা যা তরকিব নেয় তার গোটাটাকেই। মানে, এক কথায়, গোটা ডিস্কের ফাইলেরা কী ভাবে সংগঠিত। আবার একটা ডিস্কে বা পার্টিশনে কী ভাবে ফাইল রাখা হবে তারও নানা আলাদা আলাদা উপায় আছে। এই আলাদা আলাদা উপায়গুলোকেও আমরা ফাইলসিস্টেম বলে ডেকে থাকি। যেমন আমার দুটো হার্ড ডিস্কের সাতটা পার্টিশনে আছে চার রকমের ফাইলসিস্টেম, এক্সএফএস, রাইজারএফএস, সোয়াপ, আর উইন্ডোজ ফ্যাট৩২। কোথায় কোনটাকে পাব সেটা নিজের ইচ্ছেমতন, যাকে বলে মাউন্ট করা, পরে আসছি সেসবে। এবার দেখুন, এই ফাইল রাখার রাইজারএফএস, এক্সএফএস, ফ্যাট৩২ — এই যে আলাদা আলাদা রকম, আরো আছে, ইএক্সটি২ ইএক্সটি৩ ইত্যাদি — এদেরও ডাকছি আলাদা আলাদা ফাইলসিস্টেম বলে, আবার যখন একটা মেশিনের একটা সিস্টেমের গোটা ফাইল হায়েরার্কি কাঠামোটাকে ভাবছি, সেটাকেও ডাকছি ফাইলসিস্টেম বলে। আমার মেশিনে উইন্ডোজ ফ্যাট৩২ পার্টিশনদুটোকে পাওয়া যায় /mnt/windows/c এবং /mnt/windows/d ডিরেক্টরিতে, একটা রাইজারএফএস পার্টিশনকে পাওয়া যায় /mnt/slackware ডিরেক্টরিতে, অন্যটাকে /mnt/arkive ডিরেক্টরিতে, আগের সেকশনের ছবির সঙ্গে মিলিয়ে নিন। আর আমার মেশিনে মূল সুজে ৮.২ ফাইলসিস্টেমটা আছে তিনটে এক্সএফএস ফাইলসিস্টেম পার্টিশন মিলিয়ে। এই শেষ বাক্যটায় দেখুন, দুবার 'ফাইলসিস্টেম' শব্দটা ব্যবহার হয়েছে দুটো আলাদা অর্থে। সুজে ৮.২ ফাইলসিস্টেম বলতে বোঝাচ্ছি গোটা হায়েরার্কিকাল ডিরেক্টরি কাঠামোটাকে যার মধ্যে রয়েছে প্রতিটি পার্টিশনের প্রতিটি ফাইল, দ্বিতীয়বার ফাইলসিস্টেম বলতে বোঝাচ্ছি আলাদা আলাদা ধরনের পার্টিশনে আলাদা আলাদা রকমে ফাইল রাখার কায়দাকে।

একটা পার্টিশন বলতে আমরা বুঝি কাঁচা ভৌত হার্ড ডিস্কের একটা অংশকে। আর একটা ফাইলসিস্টেমকে বানিয়ে তোলা হয় এরকম এক বা একাধিক পার্টিশনকে মিলিয়ে। যেমন আমার সুজে অপারেটিং সিস্টেমের ফাইল সিস্টেম গড়ে উঠেছে সব কটা পার্টিশনকে মিলিয়ে। আবার উইন্ডোজ অপারেটিং সিস্টেমের কাছে গোটা ফাইলসিস্টেমটা হল শুধু দুটো পার্টিশনের, তাই উইন্ডোজ দেখায় আমার মোট হার্ড ডিস্ক স্পেস ৮০ জিবি-র জায়গায় মাত্র ১৬ জিবি, কারণ লিনাক্স পার্টিশনগুলোকে ও দেখতেই পায়না। এই পার্টিশনগুলোকে সরাসরি আমরা চিনি এদের ডিভাইস ফাইল দিয়ে। আর তাতে ফাইল সিস্টেম আলাদা করে তৈরি করে নিতে হয়। তাই ফাইল সিস্টেম হায়েরার্কি বোঝার আগে একটু ডিভাইস এবং পার্টিশনগুলোর সম্পর্কে জানা দরকার। এই মুহূর্তে এই পুরোটা একটু হিব্রু লাগছে হয়ত, কিন্তু আজকের আলোচনা শেষ হতে হতেই আর লাগবেনা।

| পার্টিশন | সাইজ | ফাইলসিস্টেম টাইপ | মাউন্ট ভূমি |
|---|---|---|---|
| প্রথম হার্ডডিস্ক — ভৌত সাইজ চল্লিশ জিবি — /dev/hda | | | |
| /dev/hda1 | 8 | Win95 Fat32 | /mnt/windows/c |
| /dev/hda5 | 8 gb | Win95 Fat32 | /mnt/windows/d |
| /dev/hda6 | 21.5 gb | Linux Reiserfs | /mnt/slackware |
| দ্বিতীয় হার্ডডিস্ক — ভৌত সাইজ চল্লিশ জিবি — /dev/hdb | | | |
| /dev/hdb1 | 110 mb | Linux XFS | /boot |
| /dev/hdb2 | 260 mb | Linux Swap | — |
| /dev/hdb3 | 12 gb | Linux XFS | / |
| /dev/hdb5 | 25 gb | Linux Reiserfs | /mnt/arkive |

আমার মেশিনের হার্ড ডিস্কদুটোর পার্টিশনগুলোর একটা তালিকা এখানে দিচ্ছি। একটু মিলিয়ে নিই। তারপর আমরা এদের ভালো করে বোঝার চেষ্টা করব। আমার মেশিনে দুটো হার্ড ডিস্ক, /dev/hda আর /dev/hdb, আগেই বলেছি — এখন আমরা জানি এর মানে /dev নামের ডিরেক্টরিতে দুটো ডিভাইস ফাইল, যাদের নাম hda আর hdb। দুটোরই সাইজ চল্লিশ জিবি করে। এখানে তাদের পার্টিশনগুলোর তালিকাটা দিলাম। দুটোই চল্লিশ জিবির বা চল্লিশ গিগাবাইটের হার্ডডিস্ক, কিন্তু পাওয়া যায় মোটামুটি সাড়ে সাঁইত্রিশ জিবি করে, কারণ অন্যটুকু লেগে যায় ফাইলসিস্টেম থাকার কাঠামো বানাতে। এবার তালিকাটার দিকে তাকান, এর একদম বাঁদিকে আছে ডিভাইস ফাইলের নাম, যা দিয়ে সিস্টেম এই পার্টিশনটাকে চিনছে। তারপর তার সাইজ, জিবিতে মানে গিগাবাইটে আর এমবিতে মানে মেগাবাইটে। তিন নম্বর স্তম্ভে হল ফাইল সিস্টেমের নাম, চার ধরনের আছে মোট, আগেই বলেছি। এবং একদম ডানদিকে, মাউন্ট পয়েন্ট। ‘মাউন্ট পয়েন্ট’ কাকে বলে আপনি জানেন না এখনো, কিন্তু এই সাতটা মাউন্ট পয়েন্টকেই দেখুন, আপনি পরিস্কার চিনতে পারবেন এক একটা ডিরেক্টরির নামে, দেখুন, দেখানো আছে আগের সেকশনের ছবিটায়। তার মানে এইটাই, পরে আমরা আরো ভালো করে বুঝব — মাউন্ট করা হয় একটা ভৌত ডিভাইসকে, একটা পার্টিশনকে, এবং মাউন্ট করার ভূমিটা হল একটা ডিরেক্টরি।

যখন /dev/hdb5 পার্টিশনটা মাউন্ট করা আছে, তখন আমরা /mnt/arkive ডিরেক্টরিতে ঢুকে সেখানে ls মারলে তার ফাইলগুলো দেখতে পাব — আমার সমস্ত ডাউনলোড, সমস্ত সফটওয়্যার, সমস্ত এভিআই ফাইলগুলো ওখানে রাখা — বেশ লোভনীয় ডিরেক্টরি। জিএলটির গোটা সিডিটা বানানো একটা ওয়েবপেজের মত করে, সেই সিডির মোট জিনিষপত্তরও ওখানেই রাখা linux.books নামের একটা ডিরেক্টরিতে। কিন্তু। কিন্তু যখন মাউন্ট করা নেই, আপনি chdir করে / থেকে /mnt থেকে /mnt/arkive ডিরেক্টরিতে যান, এবার ls মারুন। ফক্কা। কিছুই নেই। কারণ, ওই ভৌত উপাদান, /dev/hdb5 পার্টিশনটা, যাকে মাউন্ট করা হয় ওই ডিরেক্টরিতে, সে তখন ফাইলসিস্টেমের বাইরে, তাকে মাউন্ট করা হয়নি।

দেখুন তো, এবার কি একটু শিকারের গন্ধ নাকে আসছে? মনে হচ্ছে প্রায় পেড়ে ফেলেছেন, আর কয়েক থাবার দূরত্বে মাত্র? যদি তা নাও হয়, এই অদ্ভুত লাগাটা একদমই অপরিচয় জনিত। চেনা কাউকে দিয়ে নিজের মেশিনে ইনস্টল করে নিন গ্নু-লিনাক্স, হাতের কাছে খোঁজাখুঁজি করে একদমই কাউকে না-পেলে সবচেয়ে কাছের লাগ বা জিএলটি-তে যোগাযোগ করুন, এমনকি চেষ্টা করে নিজেও করে ফেলতে পারবেন। আমাদের জিএলটি-র দুজন নিজে নিজেই করেছে প্রথমবারই। যদিও টেলিফোন করতে হয়েছে কয়েকবার, মাঝপথে নানা অসুবিধেয়। এবার ওই ইনস্টল করা সিস্টেমে একটু সময় কাটাতে থাকবেন, প্রথম প্রথম মনে হবে সবই একটা ল্যাবিরিন্থ-এর মত, গোলোকধাঁধা, তারপর, হঠাৎ একসময় সিস্টেমের যুক্তিগঠনটা আপনার মাথায় বসে যেতে শুরু করবেন, দেখবেন, কুচি কুচি ম্যুরালের মত একটা ছকে বসে যেতে শুরু করেছে আপনা থেকেই। এই বসে যাওয়ার প্রক্রিয়াটা ত্বরান্বিত করার চেষ্টায় আসুন এবার একটু ডিভাইস, পার্টিশন, ডিস্ক, মাউন্ট এগুলোকে আর একটু ভালো করে জানার চেষ্টা করা যাক।

একটা ফাইলসিস্টেমের একদম গোড়ার কয়েকটা উপাদান হল সুপারব্লক, আইনোড, ডেটা ব্লক, ডিরেক্টরি ব্লক, এবং ইনডিরেকশন ব্লক। যেগুলো এবার একটু ধরে ধরে বুঝতে হবে আমাদের। প্রাথমিক সংজ্ঞাটা জেনে রাখা যাক। সুপারব্লকে থাকে গোটা ফাইলসিস্টেম মানে ওই পার্টিশনের গোটা ডিরেক্টরি ব্যবস্থা এবং ফাইলব্যবস্থা সম্পর্কে তথ্য। তার গঠন, তার আকার, এইসব। ঠিক কী কী তথ্য এই সুপারব্লকে থাকবে, এবং ঠিক কী ভাবে থাকবে এটা বদলায় ফাইলসিস্টেম থেকে ফাইলসিস্টেমে। ধরুন, পার্টিশনটা রাইজারএফএস হলে যেভাবে থাকবে এটা, এবং যা থাকবে সেই সুপারব্লকে, এক্সএফএস হলে সেভাবে থাকবেনা, এবং সেই তথ্যও থাকবেনা। ভিফ্যাট হলে আবার আর এক রকম। ইত্যাদি। আইনোডে থাকে একটা ফাইল সম্পর্কে যাবতীয় তথ্য, শুধু ফাইলটার নাম ছাড়া। কারণ, ফাইলসিস্টেমের কাছে নামটার কোনো মানে নেই, মানে আছে আপনার মানে ব্যবহারকারীর কাছে। ডিরেক্টরি ফাইলে লেখা থাকে ফাইলের নাম এবং আইনোড নম্বর, আগেই বলেছি। যাতে আপনি কোনো ফাইলকে টানাহেঁচড়া করছেন যখন, তার নামটা লিখছেন, ডাইরেক্টরি ফাইলে তার আইনোড নম্বরটা পড়ে সিস্টেম জেনে যেতে পারে কোন ফাইলটার কথা আপনি বলছেন। ফাইলের তথ্যটা লেখা আছে কোন কোন ডেটা ব্লকে সেটাও লেখা থাকে আইনোডে। আইনোডে মাত্র কয়েকটা ডেটা ব্লকের নম্বর লেখারই জায়গা থাকে, তার চেয়েও বেশি ডেটা ব্লকের যদি দরকার পড়ে ফাইলের মোট তথ্যটা লিখতে, তাহলে আরো আরো ডেটা ব্লকের পয়েন্টার ফাইলের সঙ্গে যোগ করা হয় একটা গতিশীল বা ডায়নামিক রকমে। এই গতিশীল ভাবে কাজে লাগানো ডেটা ব্লকগুলোকেই বলে ইনডিরেক্ট ব্লক, যাদের

নম্বর ঠিকানা ইত্যাদি লেখা থাকে ফাইলসিস্টেমের ইনডিরেকশন ব্লকে। একটু ভজোকটো লাগছে হয়ত, কেটে যাবে, বাস্তব একটা ফাইলসিস্টেমকে ধরে আলোচনা শুরু হলেই।

এই শেষ প্যারাটায় গোটাটা জুড়েই যে ফাইলসিস্টেমের কথা বলা হয়েছে সেটা এক একটা পার্টিশনে এক একটা রকমের ফাইলব্যবস্থা যে ভাবে থাকে, তার বিষয়ে। এই রকমের এক বা একাধিক পার্টিশনকে মিলিয়ে তৈরি হয় একটা পূর্ণাঙ্গ ডিরেক্টরি সিস্টেমের ফাইলব্যবস্থা। একটা ফাইলব্যবস্থা একটা পার্টিশনে গ্নু-লিনাক্সে প্রায় থাকেই না, সেটা গ্নু-লিনাক্স নিরাপত্তা এবং কাজের ধারণার সঙ্গেই মেলেনা। এবার এর সঙ্গে আরো আরো অপারেটিং সিস্টেম যোগ হলে তো সেটা আর ভৌত রকমেই সম্ভব না। স্ল্যাকওয়ারের, অন্তত বুট করার জন্যে এবং একান্ত নিজস্ব কনফিগারেশন ফাইলের জন্যে, একটা সম্পূর্ণ নিজস্ব পার্টিশন লাগবেই, ডেটার পার্টিশনগুলো সে যদি শেয়ারও করে সুজের সঙ্গে। আবার উইন্ডোজ তো গ্নু-লিনাক্স পার্টিশনকে দেখতেই পাবেনা, তার মানে, উইন্ডোজ থাকলে তার নিজের পার্টিশন লাগবেই, যদি দেখতেও পেত, তাহলেও ওই বুট করার জন্যে এবং নিজের সিস্টেম কনফিগারেশন ফাইল রাখার জন্যে একটা পার্টিশন লাগতই। তার মানে ধরুন যে কটা অপারেটিং সিস্টেম আপনার আছে, অন্তত সেই কটা পার্টিশন আপনার লাগবেই।

যখন আমার শুধু উইন্ডোজ ছিল তখনই, সেই আট জিবির হার্ড ডিস্কে ছিল তিনটে পার্টিশন, তার অবশ্য একটা রক্তমাংসের কারণ ছিল, যা গ্নু-লিনাক্সে অনুপস্থিত, সেটা হল ভাইরাস। মাঝে মাঝেই শুরু হত ভাইরাসের ভুতের নেত্য, তখন আর কি, বাপ বাপ বলে একটা পার্টিশন পুরো ফরম্যাট করে দাও, অন্যটায় ফাইল এনে। আবার যদি নিজের ভালোবাসা মাস্টার বুট রেকর্ডে বা এমবিআর-এ ছড়িয়ে দেওয়ার মত রোমান্টিক ভাইরাস হয় তাহলে তো সবগুলোই ফরমাট করো, মানে কম্ম কাবাড়, গোটা হার্ডডিস্কের প্রতি কুচি তথ্য উপড়ে ফেলো। এরকম আবার একবারই হয়েছিল, তার আগের ফাইলগুলোর প্রিন্ট-আউট ছিল, মধ্যের তেইশদিনের কাজ চলে গেছিল ভাইরাসদের অস্ত্রে। মাঝে মাঝে মনে হয়ন, কোনো ক্রিয়াই তো অবশেষহীন হয়না, আর কম্পিউটারে লেখা মানেও তো একটা ভৌত ক্রিয়া, সেই ফাইলগুলো, তার তথ্যগুলো, গেল কোথায়, কোন শূন্যতায়? এটাই কি পরম ব্রহ্মে লীন হয়ে যাওয়া, যেখানে আছে, কিন্তু আর ফিরিয়ে আনা সম্ভব না।

এই যে দেখুন, দুবার বললাম 'ফরম্যাট' করার কথা, এখানে বলেছি তথ্যশূণ্য করে দেওয়া অর্থে, আমরা সবাই করি, একটা কাঁচা হার্ড ডিস্ক কিনে এনে সেটাকে সরাসরি ব্যবহার করা যায়না। তাকে ফাইল রাখার, তথ্য রাখার, পরে সেই তথ্য ব্যবহার করার মত করে একটা আকার দিতে হয়, মানে একটা ফাইলসিস্টেমের কঙ্কাল বানাতে হয়। শিখিয়ে পড়িয়ে নিতে হয়, সদ্য দলে আসা রাজনৈতিক কর্মীকে যেমন নেতা হওয়ার আগে, অনেক যত্নে কেয়ারে তিরস্কারে, আগে সভ্য এবং চোর বানিয়ে তুলতে হয়। কোথায় কোন জিনিসের কী মানে এটা সে নিজে থেকেই যাতে বুঝে যেতে পারে। পুরোনো মানে দিয়ে যাতে আর না-বোঝে। গ্নূ-লিনাক্সে এটা করা হয় মেক-ফাইলসিস্টেম (mkfs) দিয়ে। এর পুরোটারই একটা ছোট্ট অংশ হল আগে ভৌত হার্ড ডিস্কটাকে ইলেকট্রনিক তথ্যের থেকে মুক্ত করা।

৭।। ডিভাইস, ডিভাইস-ফাইল, ডিভাইস-নাম

আমরা জানি গ্নু-লিনাক্সে সব ডিভাইসই এক একটা ফাইল যারা /dev ডিরেক্টরিতে বা /dev ডিরেক্টরির পেটের মধ্যে আরো আরো সাবডিরেক্টরিতে আছে। যত দিন বেড়েছে, এই /dev ডিরেক্টরির আকার এবং জটিলতাও বেড়েছে, কারণ প্রকৌশল বদলেছে, আরো আরো নতুন ধরনের ডিভাইস আবিষ্কার হয়েছে, তাদের নিয়ে আসা হয়েছে ব্যবহারে, অপারেটিং সিস্টেমের মধ্যে। একটা ডিভাইস খোলা, ব্যবহার করা বন্ধ করা মানে, ওই ডিভাইসের ফাইলটাকে খোলা, পড়া, লেখা, বন্ধ করা। ডিভাইস পড়তে বললেই কী করে করতে হবে, এবং কী করতে হবে, কোথায় কোন ড্রাইভারকে কী ভাবে কাজে লাগাতে হবে, এটা কারনেল জানে, সেটা জানে বলেই সে কারনেল। দুই নম্বর দিনের আলোচনাগুলো মনে করুন। আজকের সেকশন ৪-এর শেষ ছবিটায় দেখানো বিভিন্ন সিস্টেম ডিরেক্টরিগুলোর আলোচনায় আমরা আসছি একটু পরেই, গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির আলোচনায়, কোন ডিরেক্টরিতে কী থাকে এবং কেন থাকে গ্নু-লিনাক্সে তার চালু প্রথায়। এখানে কয়েকটা ডিভাইস ফাইলের সূত্রে /dev ডিরেক্টরির কথা একটু বলে নেওয়া যাক। এই ডিভাইস ফাইলগুলো একটা গড় উদাহরণ, এক একটা সিস্টেমে এর থেকে আলাদা হতেই পারে, হয়েই থাকে, আমরা ছকটা বোঝার চেষ্টা করছি।

/dev ডিরেক্টরিতে যে ফাইলগুলো থাকে তাদের মধ্যে আছে হার্ড ডিস্ক — /dev/hd, আমরা আগেই দেখেছি, দুটো হার্ড ডিস্ক থাকলে হয় /dev/hda এবং /dev/hdb। ধরুন, উইনডোজ হলে, যদি দুটো হার্ড ডিস্কেই একটা করে পার্টিশন থাকে তাহলে হত C:\ এবং D:\। আবার দুটো হার্ড ডিস্কেই দুটো করে পার্টিশন থাকলে হত C:\ D:\ E:\ F:\ ইত্যাদি। /dev/hda এবং /dev/hdb দুটোই আইডিই হার্ড ডিস্ক। আগেই তো বলেছি, আইডিই মানে ইন্টিগ্রেটেড ডিভাইস ইলেকট্রনিক্স, এক ধরনের ডিস্ক ড্রাইভ যাতে ডিভাইসের ড্রাইভারটা দেওয়া থাকে ড্রাইভের মধ্যেই, আলাদা করে অ্যাডাপ্টার কার্ড আর লাগাতে হয়না। এর থেকে উন্নত এবং তাই দামী হার্ড ডিস্ক হল স্কাসি, স্মল কম্পিউটার সিস্টেম ইন্টারফেস। আইডিই হার্ড ডিস্কের থেকে অনেক দ্রুত গতিতে তথ্য পড়া এবং লেখা যায় স্কাসি ড্রাইভে। একটা বিশেষ ধরনের ব্যাবস্থা এই স্কাসি। এতে যে শুধু হার্ড ডিস্ক হয় তা নয়, অন্য পেরিফেরালও হয়। স্কাসি হার্ড ডিস্ক প্রথমটার নাম হয় /dev/sda, দ্বিতীয়টার নাম হয় /dev/sdb। এবার আইডিই বা

স্কাসি দুরকম হার্ড ডিস্কেই একাধিক করে পার্টিশন থাকতে পারে, /dev/hdb1 মানে দ্বিতীয় আইডিই হার্ড ডিস্কের এক নম্বর পার্টিশন, /dev/sda3 মানে প্রথম স্কাসি হার্ড ডিস্কের তিন নম্বর পার্টিশন, ইত্যাদি। আইডিই সিডি ড্রাইভ হলে তার ডিভাইস ফাইল হয় /dev/sr0, আর একটা থাকলে /dev/sr1, এখানে শূন্য মানে প্রাইমারি, এক মানে স্লেভ। এদের লিংক বা সংযোগ করা থাকে /dev/cdrom0, /dev/cdrom1 ইত্যাদি নানা ফাইল দিয়ে। সংযোগ করা বলতে কী বোঝায় সেকথায় আমরা পরে আসব। স্কাসি সিডি ড্রাইভ হলে তখন নাম হয় /dev/scd। একাধিক থাকলে /dev/scd0, /dev/scd1। শুধু হার্ড ডিস্ক নয়, সিডি রম, মেমরি এরকম অনেক কিছুই স্কাসি হতে পারে। উইনডোজে যাকে কমিউনিকেশন পোর্ট কম-ওয়ান (COM1) বলে, গ্নু-লিনাক্স তাকে ডাকে /dev/ttys0 বলে। মানে প্রথম সিরিয়াল পোর্ট। সচরাচর এতে মোডেম লাগানো থাকে। তখন /dev/ttys0 আর /dev/modem পরস্পর সংযুক্ত করা থাকে। একটা বললেই অন্যটাকে বোঝায়। /dev/modem থেকে আমরা তথ্য পড়ছি মানে /dev/ttys0 পোর্টে যুক্ত মোডেম থেকে কোনো মেল বা ওয়েবপেজ নামাচ্ছি মেশিনে। ইউএসবি (USB — **U**niversal-**S**erial-**B**us) ডিভাইসগুলোর ফাইলগুলো থাকে /dev/usb ডিরেক্টরিতে। ইউনিভার্সাল সিরিয়াল বাস হল একটা উন্নত বাস-ব্যবস্থা, বাস মানে তো আমরা জানি, পেরিফেরালগুলোর সঙ্গে যোগাযোগপথ, মোটামুটি '৯৮-এর পর থেকে ব্যবহৃত হচ্ছে। পিএসবাইটু (PS/2) পোর্টের ডিভাইস ফাইল হয় /dev/psaux। এই পিএসটু এসেছে আইবিএম-এর একটা পিসি মডেলের নাম থেকে, পিসিটা খুব একটা চলেনি, অন্য ধরনের একটা নির্মাণের কারণে, কিন্তু নামটা রয়ে গেছে। পিএসবাইটু পোর্টে মাউস, মোডেম ইত্যাদি ব্যবহার হয়। ফ্লপি ড্রাইভ সচরাচর হয় /dev/fd0। /dev/dsp মানে তো আগেই বলেছি, সাউন্ড কার্ড বা প্রথম সাউন্ড ডিভাইস। /dev/lp0 মানে প্রিন্টার পোর্ট বা এলপিটি-ওয়ান (LPT1)। এই যে নামকরণগুলো লিখলাম, ডিরেক্টরিতে দেখবেন, কখনো কখনো তার একটা যৌক্তিক তালমিল পাওয়া যায় বটে, কিন্তু বেশিরভাগ কেসেই বেশ ইচ্ছেমাফিক মর্জিমাফিক।

এই ডিভাইসগুলোর মধ্যে আবার ব্লক ডিভাইস আর ক্যারেকটার ডিভাইস এই দুই ভাগ আছে, আমরা আগেই বলেছি, তার সঙ্গে মিলিয়ে ফাইলগুলোও ব্লক ডিভাইস ফাইল আর ক্যারেকটার ডিভাইস ফাইল। যেমন, ফ্লপি, হার্ড ডিস্ক, সিডি ড্রাইভ — এগুলো ব্লক ডিভাইস, আবার সিরিয়াল পোর্ট, মাউস, প্যারালাল প্রিন্টার পোর্ট — এগুলো ক্যারেকটার ডিভাইস। ব্লক ডিভাইসে তথ্য লেখা এবং পড়াটা হয় ব্লকের এককে। একটা প্রোগ্রাম যখন একটা ফাইলকে লিখতে চায় একটা ব্লক ডিভাইসে, সে আবেদন জানায় কারনেলের কাছে, কারনেল তখন ডিভাইস ড্রাইভারকে কাজে লাগিয়ে এক এক তাল তথ্য, মানে এক এক ব্লক করে লিখতে থাকে ডিস্কের শরীরে থাকা ওই ফাইলে, প্রতিটি ব্লকের একটা নিজের নম্বর আছে, দুই নম্বর আর এক নম্বর দিনের আলোচনা থেকে মনে করুন। আবার ডিস্ক থেকে একটা ফাইল পড়ছি যখন, প্রথমে খোঁজা হয় বাফার ক্যাশে, মানে সেই তুলে রাখা অস্থায়ী স্মৃতি, সেখানে সাম্প্রতিকতম অতীতে ব্যবহৃত তথ্যের মধ্যে পাওয়া যায় কিনা। যদি মিলল তো গুড, আর ডিস্ক নাড়াচাড়া করতে হলনা, অনেকটা সময় বেঁচে গেল, একটু আগে গ্নু-লিনাক্স ব্যবস্থার মেমরি ব্যবহারের যে বাড়তি ওস্তাদিটার কথা বলছিলাম।

এবার আমাদের শিখে যাওয়া একটা কমান্ড একটু ব্যবহার করা যাক /dev ডিরেক্টরির দুচারটে ফাইলের উপর। আমরা /dev ডিরেক্টরিতে গিয়ে যদি '`ls -l -R`' মারি, তাহলে একটা বিরাট সময় ধরে অজস্র ফাইলের লং লিস্টিং বা দীর্ঘ তালিকা স্ক্রিন জুড়ে দেখিয়ে যাবে সিস্টেম, যেখানে একটা ফাইলের প্রচুর খুঁটিনাটি ফুটিয়ে তুলবে সিস্টেম। আমরা কিন্তু এখানে '`ls -lR`' কমান্ডও মারতে পারতাম, '`ls`' কমান্ড তার অপশানগুলোকে এভাবে জুড়ে দিতে দেয়, যদিও সব কমান্ড দেয়না। কোনটা দেয়, কোনটা দেয়না এটা আপনাকে নিজেকে শিখতে হবে। গ্নু-লিনাক্স কমিউনিটির অজস্র মানুষ মিলে শুধু আপনি সেখান থেকে পড়ে শিখবেন বলে ওই বিপুল পরিমাণ টেক্সট লিখে রেখেছে, যা আপনার সিস্টেমে এমনিতেই দেওয়া থাকে। এখানে এই '`l`' অপশানটা বলে ফাইলের সমস্ত প্রয়োজনীয় খুঁটিনাটিকে লং বা দীর্ঘ আকারে ফুটিয়ে তুলতে, আর '`R`' অপশানটা বলে ওই ডিরেক্টরির মধ্যে যে সাবডিরেক্টরিগুলো আছে তাদের মধ্যের ফাইলগুলোকেও দেখাতে। এই শেষহীন তালিকাটাকে রিডাইরেক্ট করে একটা ফাইল বানাতেও জানি আমরা। সেই ভাবে তৈরি করা ফাইল থেকে কয়েকটা লাইন তুলে দিই, আদতে ফাইলটায় লাইনের সংখ্যা এগারো হাজারের মত। আসলে যা যা ঘটতে পারে সব সম্ভাবনাই তৈরি থাকে, যেমন আটখানা হার্ড ডিস্কের প্রতিটায় যাতে একত্রিশটা অব্দি পার্টিশন বানানো যায়, hda1 থেকে hda31, hda থেকে hdh, আরো বারোখানা হার্ড ডিস্কে যাতে পনেরোটা অব্দি পার্টিশন বানানো যায়, hdi1 থেকে hdi15, hdi থেকে hdt, সেই ব্যবস্থাও করে রাখা আছে আগে থেকেই। এবং এটা সব রকম ডিভাইসের জন্যেই। এই ডিভাইসাড়ম্বরের একটা যুক্তি নিশ্চয়ই আছে, আমি জানিনা, ভালো করে কম্পিউটার জানে এমন কারো কাছে জিগেশ করতে হবে।

```
lrwxrwxrwx  1 root  root       10 2003-12-05 14:53 modem -> /dev/ttyS0
crw-rw----  1 root  uucp   4,  64 2003-03-14 18:37 ttyS0
crw-rw--w-  1 root  lp     6,   0 2003-03-14 18:37 lp0
brw-rw----  1 root  disk   3,   0 2003-03-14 18:37 hda
brw-rw----  1 root  disk   3,   1 2003-03-14 18:37 hda1
brw-rw----  1 root  disk   3,  64 2003-03-14 18:37 hdb
```

```
brw-rw----  1 root  disk   3,  65 2003-03-14 18:37 hdb1
lrwxrwxrwx  1 root  root       10 2003-12-05 07:09 mouse -> /dev/psaux
crw-rw----  1 root  root   10,  1 2003-12-17 11:17 psaux
lrwxrwxrwx  1 root  root        8 2003-12-05 07:38 cdrom -> /dev/sr1
lrwxrwxrwx  1 root  root        8 2003-12-05 07:38 cdwri -> /dev/sr0
brw-r--r--  1 root  disk   11,  0 2003-03-14 18:37 sr0
brw-r--r--  1 root  disk   11,  1 2003-03-14 18:37 sr1
brw-r--r--  1 root  disk   11,  0 2003-03-14 18:37 scd0
brw-r--r--  1 root  disk   11,  1 2003-03-14 18:37 scd1
```

এবার দেখুন তো কয়েকটা ফাইলের এই লং লিস্টিং থেকে আপনি কিছু আন্দাজ করতে পারছেন কিনা? ঠিক এই আগের দুতিনটে প্যারাগ্রাফে যা আমরা বলে এলাম, সেইগুলোকে? প্রতিটি ফাইলের নামের আগেই সিস্টেমের কাছে তার অ্যাবসলিউট বা চূড়ান্ত ঠিকানা '/dev/' যোগ করে নিন, কারণ, এই 'ls' কমান্ডটা তো আমরা দিয়েছি '/dev' ডিরেক্টরিতে দাঁড়িয়েই।শুধু একটা জিনিষ মনে রাখুন, লাইনের একদম গোড়ায় যে 'b' বা 'c' — তার মানে ব্লক ডিভাইস বা ক্যারেকটার ডিভাইস, আর 'l' মানে লিংক, ওই যে সংযোগের কথা বললাম, পরে আরো ভালো করে জানব। এবার একদম ডানদিকের ডিভাইসের নামের সঙ্গে মেলাতে পারছেন? চারটে ডিভাইসের নামের পরে দেখুন '->' চিহ্ন দিয়ে অন্য একটা ডিভাইস দেখানো, এর মানে মুখ আর মুখোশ। যেমন '/dev/psaux' ডিভাইসের মুখোশ হল '/dev/mouse'। কোনো প্রোগ্রাম '/dev/mouse' বললে সিস্টেম আসলে '/dev/psaux' বুঝবে। আমরা 'cat /dev/mouse' করে মাউসের নড়াচড়ার টেক্সট সংকেত স্ক্রিনে দেখেছিলাম, মনে পড়ছে? এই মেশিনে মাউস ডিভাইসের ফাইলটা '/dev/psaux' হল কারণ এটা পিএসবাইটু মাউস। এই ডিভাইস ফাইলগুলো এবং বাস্তব ভৌত উপাদানগুলো সঠিক অর্থেই তুল্যমূল্য, আমরা আগেই, বিভিন্ন ডিভাইস ফাইলগুলো নিয়ে মজা করার সূত্রে দেখিয়েছি, কী ভাবে একদম ভৌত উপাদানগুলোর মতই এই ডিভাইসগুলোয় তথ্য লেখা যায়, বা এখান থেকে তথ্য পড়া যায় — মাউস-চলনের ফরমুলা বা কারনেলের কণ্ঠস্বর তারই প্রমাণ।

ডিভাইসদের একটা ভাগ যেমন ব্লক আর ক্যারেকটার, তেমনি আর একটা ভাগ হল মেজর আর মাইনর নাম্বার, মুখ্য আর গৌণ সংখ্যা। মেজর নাম্বার বা মুখ্য সংখ্যা দিয়ে বোঝায় ওই ধরনের ডিভাইসের বর্গ, আর মাইনর নাম্বার বা গৌণ সংখ্যা দিয়ে বোঝায় ডিভাইসটার নিজস্ব বর্গ। ডিভাইসটা মাস্টার না স্লেভ, বা কত নম্বর পার্টিশন এগুলো বোঝায় এই গৌণ সংখ্যা দিয়ে। ফাইলের ওই দীর্ঘ তালিকাটায় দেখুন পাঁচ নম্বর স্তম্ভটা হল মেজর আর মাইনর নাম্বারের। যেমন 'hda1' ডিভাইসের সংখ্যাদুটো হল '3, 1', আবার 'hdb1' ডিভাইসের সংখ্যাদুটো '3, 65'। একই ডিভাইস কন্ট্রোলার তাদের দুজনকেই নিয়ন্ত্রণ করে বলে এদের দুজনেরই মেজর নাম্বার সমান। এই জায়গাটা নিয়ে আমার আর না-এগোনোই ভালো, আমি নিজেও এইরকম ভাসাভাসা ভাবেই জানি। ভালো করে জানতে চাইলে কোলকাতা লাগে আমাদের মেইলিং লিস্ট আছে কী করতে? আপনি কত চান? পড়ে আর কূল পাবেন না। জিসিসিতে math.h বলে একটা হেডার ফাইল আমি, নিজের বুদ্ধির দোষেই, ভালো করে ম্যানুয়াল পড়িনি, কিছুতেই কাজে লাগাতে পারছিলাম না, লিস্টে জানানোর দু-ঘন্টার মধ্যে গোটা পাঁচেক মেল এসেছিল, কৌশিক ঘোষ বলে একজন ধরে ধরে কমান্ড তুলে তুলে আমায় বুঝিয়ে দেয়, এইসব ছেলেরা যা ভালো জানে সব — সে আর বলার না।

কিন্তু এখানে একটা সমস্যা তৈরি হচ্ছে, এই ডিভাইস ফাইলগুলো নিয়ে কথা বলার সময় দেখছিলাম, কিছু কথা বোঝানো যাচ্ছেনা, মানবসমাজ ও সভ্যতার আদিমতম সমস্যা মালিকানা এবং অধিকার নিয়ে কিছু কথা না-বলে। কম্পিউটারের ফাইল ব্রহ্মাণ্ডের ভূগোল নিয়ে কথা বলা শেষ করার আগে অব্দি তার এথিকস নিয়ে কিছু শুরু করব না, ঠিক করেছিলাম, কিন্তু সেটা দেখছি হচ্ছেনা, ফাইলের সম্পত্তিসম্পর্ক নিয়ে একটা সেকশন নামিয়ে নেওয়া যাক। আসলে এই বইটা লেখার আগে এই বইটা তো কোনোদিন লিখিনি, কিছু কিছু জায়গা ঘেঁটে যাচ্ছে। যাকগে, আপনারাও তো এটা পড়ার আগে কোনোদিন এটা পড়েনইনি, পড়ার পরেও পড়বেন বলে মনে হয়না। কিন্তু, আজকের আলোচনা এই অব্দিই থাক। এর পরের আলোচনা পরের দিন।

আবার আমাদের প্ল্যান বদলালো। আজকে শেষ হলনা ফাইলসিস্টেম। আসলে লেখার আগে যেভাবে ভেবেছিলাম, লেখার সময়ে দেখছি তার থেকে বদলে যাচ্ছে গোটাটা। অনেক কথা মাঝপথে বলে নিতে হচ্ছে, যা অন্যভাবে বলার কথা ভেবেছিলাম। এর পরের দিনটাও গোটাটাই এই ফাইলসিস্টেমে চলে যাবে বলেই মনে হচ্ছে। আমরা শুরু করব ফাইলের অনুমতি মালিকানা নিরাপত্তা নিয়ে। তারপর যাব একটা ভৌত পার্টিশনে কী করে ফাইল সিস্টেম লেখা থাকে সেই কথায়। তারপর শেষ করব গ্নু-লিনাক্স ফাইল সিস্টেম হায়েরার্কি দিয়ে। এখন লেখাটাও এগোচ্ছে অত্যন্ত বিরক্তিকর রকমের শ্লথগতিতে। আরো এখন তো কলেজ চলছে। রোজ কলেজ করে আর কতটুকু সময়ই বা পাচ্ছি লেখার? দেখুন না, আজ একুশে ডিসেম্বর, তাও তো এটায় যা যা

রাখার প্ল্যান ছিল প্রথমে তার অনেকটাই রাখতে পারলাম না। পরের দিন করতে হবে। ফাইলসিস্টেম হায়েরার্কি শেষ হয়ে যাবে আজ এরকমই ভেবেছিলাম। নিজেকেও অনেক পড়তে হচ্ছে। জিএলটি ইশকুলের ক্লাসের চেয়ে এখানে পরিশ্রম অনেক বেশি হচ্ছে, বলার সময়ে যা চট করে বলে চলে যাওয়া যায়, নিজেরও খেয়াল পড়েনা, দুটো চারটে করে ডিটেইলস-এর ফাঁক রয়ে যাচ্ছে, ঠিক নিজে যে ভাবে ভাবি সেভাবেই তো বলি। কিন্তু এখানে লিখতে গিয়ে অনেক খুচরো পাপের প্রায়শ্চিত্ত করতে হচ্ছে। আমার মাথায় নয় করার সময় এই কৌতূহলটা আসেনি, অন্য আর একজনের তো আসতেই পারে, তার তো কাজে আসবে না তখন টেক্সটটা। আর আপনিও সক্রিয় হয়ে উঠুন গ্নু-লিনাক্সে, এবং তার কমান্ড মোডেই, উইনডোজ-এর মত গুই দিয়ে নয়, সেটাই তো এই বইটা চায়। শেখানোটা বড় নয়, সেই কাজে আমি খুব দড়ও নই, নিজেই তো শিখছি একটু একটু করে, বড় কথা আপনার চিন্তা আর কৌতূহলকে উত্তেজিত করা। সেটা পারছি কিনা কে জানে? লিখি তো সবসময়েই, কিন্তু এই লেখাটা আমার কাছেও খুব ভিন্ন — নিজের এলাকার বাইরে একটা টেকনিকাল আন্দোলনের গতিটাকে হাজির করা আমারই মত অটেকনিকাল মানুষদের কাছে। যারা ঠিক মত টেকনিকাল রকমে কম্পিউটার শেখেন তাদের জন্যে তো এই বই নয়।

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত